DOCUMENT RESUME

ED 140 853 IR 004 982

AUTHOR	Duffy, Larry R.; And Others
TITLE	Design of Training Systems: Computerization of the Educational Technology Assessment Model (ETAM). Volume 2.
INSTITUTION	International Business Machines Corp., Cape Canaveral, Fla.
SPONS AGENCY	Naval Training Equipment Center, Orlando, Fla. Training Analysis and Evaluation Group.
REPORT NO	TAEG-R-40
PUB DATE	May 77
CONTRACT	N61339-73-C-0097
NOTE	380p.
EDRS PRICE	MF-$0.83 HC-$20.75 Plus Postage.
DESCRIPTORS	Computer Oriented Programs; *Decision Making; *Educational Assessment; Educational Innovation; Educational Programs; *Educational Technology; Management Systems; Military Training; *Models; Summative Evaluation
IDENTIFIERS	Educational Technology Assessment Model

ABSTRACT

This is the appendix of the final report (IR 004 981) which summarizes the analysis, design and development activities associated with the Educational Technology Assessment Model (ETAM). It includes the results of a study on the indexing of innovations and the assignment of taxonomic descriptors to courses, job/tasks, and instructional vehicles. It also includes program documentation on the ETAM Range-of-Effect, bibliographic references, and additional information supporting the ETAM design. (Author/DAG)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

TAEG REPORT NO. 40

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

ED140853

DESIGN OF TRAINING SYSTEMS

COMPUTERIZATION OF THE EDUCATIONAL TECHNOLOGY ASSESSMENT MODEL (ETAM)

VOLUME 2

Robert B. Miller

Larry R. Duffy

James D. Staley

BEST COPY AVAILABLE

This Study Was Performed by

International Business Machines Corporation

for the

Training Analysis and Evaluation Group

May 1977

GOVERNMENT RIGHTS IN DATA STATEMENT

Reproduction of this publication in whole or in part is permitted for any purpose of the United States Government.

Alfred F. Smode

ALFRED F. SMODE, Ph.D., Director
Training Analysis and Evaluation Group

IR 004 982

FOREWORD

The Design of Training Systems (DOTS) project objectives are in consonance with the requirements of Advanced Development Objective ZPN07 (formerly ADO 43-03X), Education and Training Development. ZPN07 includes a number of projects concerned with demonstrating and evaluating the technical, operational and financial feasibility of applying advanced technological applications to improving the training process.

The Bureau of Naval Personnel initiated the original ADO in 1966 to make naval training more responsive to the changing times. As one project under this effort, DOTS was designed to improve the process of managing training resources through application of the techniques of system analysis and system simulation as accomplished through mathematical modeling. The end objective is a family of computerized mathematical models enabling training management to more rapidly predict the impact of changes in training resource availability or requirements.

The majority of education and training was reorganized in 1971 under one command, Chief of Naval Education and Training (CNET). Because of this change, DOTS responsibility was transferred to CNET in March of 1972; more specifically, to the Training Analysis and Evaluation Group (TAEG), Orlando, Florida. The new CNET organization greatly increased the potential benefits to be gained from the increased application of new management techniques and, therefore, from the DOTS' R&D effort. DOTS began in February of 1973, with the majority of tasking being contracted to the International Business Machines Corporation, Federal Systems Division, Cape Kennedy Facility, located at Cape Canaveral, Florida.

The Training Analysis and Evaluation Group, Dr. A. Smode, Director, project team members Messrs. M. Middleton and W. Lindahl, complemented the contracted effort by providing direction and guidance and in establishing organizational interfaces.

The model was developed by Mr. L. R. Duffy, Dr. R. B. Miller, and Mr. J. D. Staley. Mr. R. E. Hallman and Mr. L. R. Duffy provided management for the project.

Technical Report: TAEG REPORT NO. 40

DESIGN OF TRAINING SYSTEMS
COMPUTERIZATION OF THE EDUCATIONAL TECHNOLOGY ASSESSMENT MODEL (ETAM)
VOLUMES 1 AND 2

ABSTRACT

This two volume final report summarizes the analysis, design and development activities associated with the Educational Technology Assessment Model (ETAM). It contains relevant background information and results of prior studies leading to the finalized ETAM procedures and computerized routines. A comparison of the manual versus the automated approach is included. Data base structures and the ETAM program flow are described and related to each other. The appendix includes the results of a study on the indexing of innovations and the assignment of taxonomic descriptors to courses, job/tasks, and instructional vehicles. The appendix also includes program documentation on the ETAM Range-of-Effect, bibliographic references, and additional information supporting the ETAM design. The detailed ETAM procedures supporting this report are included in TAEG Report No. 12-3, Phase II-A Report. TAEG Report No. 32, The Development of Scaling Procedures, supports the computerized approach taken in scaling variables. Other important background and reference information can also be found in a report written by Drs. R. B. Miller and A. F. Smode titled "Major Innovations in Training Technology."

Phases I, II, II-A, III and IV were accomplished by the IBM Federal Systems Division with the Training Analysis and Evaluation Group, Orlando, Florida, providing technical guidance and support. The overall DOTS objective is to provide Naval Education and Training Command (NAVEDTRACOM) management with additional tools in the form of computerized mathematical models to assist in predicting the quantitative impact of training resource decisions. The planning process will be enhanced by providing decision makers with the capability to economically and rapidly consider a wider range of alternatives.

Phase I was a study and definition effort resulting in a complete functional description of the NAVEDTRACOM; a strategic definition of the social, political, economic and technological environments pertinent to the naval education and training system in the 1980's; a list of existing and potential models amenable to computerization and to improving the decision-making process.

Phase II was devoted to the selection and development of three mathematical models from the Phase I list of candidates. The three were the System Capabilities/Requirements and Resources (SCRR), the Individualized Training Simulation System (ITSS), and the Training Process Flow (TPF) models.

Phase III centered on evaluating the selected models at the Fleet Training Center, Norfolk, VA. An important recommendation from the Test and Evaluation conducted during Phase III was that DOTS should investigate model applications at higher command levels.

Phase IV responded to the recommendations of Phase III by (1) operating and testing at TRAPAC, San Diego, CA, the models developed in Phase II, and (2) developing a new Training Requirements Analysis Model (TRAM) and field testing it at CNTECHTRA, Memphis, Tenn.

The major effort supporting Phase II-B was the ETAM procedural development during Phase II-A. In addition to the procedural development, computer applications were described, scenarios using the ETAM procedures were prepared, and ETAM validity and logical structure were confirmed.

Reproduction of this publication in whole or in part is permitted for any purpose of the United States Government.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TAEG Report No. 40	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DESIGN OF TRAINING SYSTEMS Computerization of the Educational Technology Assessment Model (ETAM) - Volume 2		5. TYPE OF REPORT & PERIOD COVERED Final 8/76 - 5/77
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Larry R. Duffy Robert B. Miller, Ph.D. James D. Staley		8. CONTRACT OR GRANT NUMBER(s) N61339-73-C-0097
9. PERFORMING ORGANIZATION NAME AND ADDRESS International Business Machines 7900 N. Astronaut Blvd. Cape Canaveral, FL 32920		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Training Equipment Center Training Analysis and Evaluation Group Orlando, FL		12. REPORT DATE May 1977
		13. NUMBER OF PAGES 344
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release:
Distribution is Unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Training Management Systems	Training Technology
Computer Based Models	Interactive Training Data Base
Training Resources Evaluation Modeling	Decision Theory
Educational Technology	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This two volume final report summarizes the analysis, design and development activities associated with the Educational Technology Assessment Model (ETAM). It contains relevant background information and results of prior studies leading to the finalized ETAM procedures and computerized routines. A comparison of the manual versus the automated approach is included. Data base structures and the ETAM program flow are described and related to each other. The appendix includes the results of a study on the indexing of innovations and the assignment of taxonomic descriptors to courses, job/tasks, and instructional vehicles.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Cont'd)

The appendix also includes program documentation on the ETAM Range-of-Effect, bibliographic references, and additional information supporting the ETAM design. The detailed ETAM procedures supporting this report are included in TAEG Report No. 12-3, Phase II-A Report. TAEG Report No. 32, The Development of Scaling Procedures, supports the computerized approach taken in scaling variables. Other important background and reference information can also be found in a report written by Drs. R. B. Miller and A. F. Smode titled "Major Innovations in Training Technology."

Phases I, II, II-A, III and IV were accomplished by the IBM Federal Systems Division with the Training Analysis and Evaluation Group, Orlando, Florida, providing technical guidance and support. The overall DOTS objective is to provide Naval Education and Training Command (NAVEDTRACOM) management with additional tools in the form of computerized mathematical models to assist in predicting the quantitative impact of training resource decisions. The planning process will be enhanced by providing decision makers with the capability to economically and rapidly consider a wider range of alternatives.

Phase I was a study and definition effort resulting in a complete functional description of the NAVEDTRACOM; a strategic definition of the social, political, economic and technological environments pertinent to the naval education and training system in the 1980's; a list of existing and potential models amenable to computerization and to improving the decision-making process.

Phase II was devoted to the selection and development of three mathematical models from the Phase I list of candidates. The three were the System Capabilities/Requirements and Resources (SCRR), the Individualized Training Simulation System (ITSS), and the Training Process Flow (TPF) models.

Phase III centered on evaluating the selected models at the Fleet Training Center, Norfolk, VA. An important recommendation from the Test and Evaluation conducted during Phase III was that DOTS should investigate model applications at higher command levels.

Phase IV responded to the recommendations of Phase III by (1) operating and testing at TRAPAC, San Diego, CA, the models developed in Phase II, and (2) developing a new Training Requirements Analysis Model (TRAM) and field testing it at CNTECHTRA, Memphis, Tenn.

The major effort supporting Phase II-B was the ETAM procedural development during Phase II-A. In addition to the procedural development, computer applications were described, scenarios using the ETAM procedures were prepared, and ETAM validity and logical structure were confirmed.

TABLE OF CONTENTS

APPENDIX PAGE

A STUDY OF DESCRIPTOR AND INDEXING TECHNIQUES FOR ETAM RANGE-OF-EFFECT ANALYSIS

INTRODUCTION A-1

PROCEDURES FOR INDEXING THE INNOVATION A-2

PROCEDURES FOR INDEXING INSTRUCTIONAL VEHICLES, TRAINING COURSES AND JOB-TASKS A-11

INDEXING OBJECTIVES A-12

SEARCH RATIONALES AND STRATEGIES A-17

B KEY FIGURES AND ILLUSTRATIONS REFERENCED IN THIS ETAM REPORT

C TRAINING COST MODEL PROGRAM

D ETAM RANGE-OF-EFFECT PROGRAM DOCUMENTATION

SECTION D.1 INTRODUCTION D.1-1

ORGANIZATION D.1-1

EQUIPMENT ENVIRONMENT D.1-1

PROGRAM ENVIRONMENT D.1-2

ETAM SYSTEM INITIALIZATION D.1-3

LOAD OF THE ABBREVIATED COURSE DATA BASE . . D.1-3

ABBREVIATED VEHICLE DATA BASE LOAD D.1-6

ABBREVIATED TASKS DATA BASE LOAD D.1-6

ESTABLISHING A NEW ETAM PROJECT D.1-9

EXECUTION OF THE RANGE OF EFFECT SEARCH . . D.1-9

PROCESSING OF SEARCH RESULTS D.1-11

TABLE OF CONTENTS

APPENDIX

PAGE

SECTION D.2 PROGRAM DOCUMENTATION D.2-1

PROGRAM P1 . . . . D.2-2

PROGRAM P2 . . . . D.2-7

PROGRAM P3 . . . . D.2-13

PROGRAM P5A D.2-17

PROGRAM P5B D.2-22

PROGRAM P5C D.2-27

PROGRAM P7 . . . . D.2-32

PROGRAM P7A D.2-38

PROGRAM P7B D.2-41

PROGRAM P8 . . . . D.2-47

PROGRAM P8A D.2-52

PROGRAM P9 . . . . D.2-58

PROGRAM P9A D.2-62

PROGRAM P14 . . . . D.2-68

PROGRAM P16 . . . . D.2-72

PROGRAM P17 . . . . D.2-79

PROGRAM P18 . . . . D.2-84

PROGRAM P20 . . . . D.2-87

SECTION D.3 NCSS EXECUTIVE SEQUENCES ASSOCIATED WITH MULTIPLE PROGRAMS . . D.3-1

SECTION D.4 ETAM FILE FORMATS, AS IMPLEMENTED D.4-1

SECTION D.5 MISCELLANEOUS CONTROL CARD SOURCE LISTINGS D.5-1

TABLE OF CONTENTS

APPENDIX PAGE

SECTION D.6 SOURCE LISTINGS OF PL/1 PROGRAMS D.6-1

E AN INTERACTIVE COMPUTER INTERVIEW

F BIBLIOGRAPHY

LIST OF FIGURES

FIGURE NO.		PAGE
A-1	FORMAT OF DESCRIPTORS FOR INDEXING AN INNOVATION OR DATA BASE ENTITIES: TRAINING COURSES	A-4
A-2	FORMAT OF DESCRIPTORS FOR INDEXING AN INNOVATION OR DATA BASE ENTITIES: JOB TASKS	A-5
A-3	FORMAT OF DESCRIPTORS FOR INDEXING AN INNOVATION OR DATA BASE ENTITIES: INSTRUCTIONAL VEHICLES	A-6
A-4	EXAMPLES OF ENTITY SELECTION BASED UPON INNOVATION DESCRIPTORS (TRAINING COURSE DESCRIPTORS)	A-7
B-1	SUMMARIZED PRELIMINARY FEASIBILITY PROFILE	B-3
B-2	COST/SAVINGS DATA SHEET	B-4
B-3	DECISION TREE FOR ASSESSMENT OF TRAINING INNOVATION (EXAMPLE OF FORM COMPLETED WITH INITIAL OUTCOME VALUES AND PROBABILITIES)	B-5
B-4	DECISION TREE FOR ASSESSMENT OF TRAINING INNOVATION (EXAMPLE OF FOLDING BACK PROCESS)	B-6
B-5	FORMAT FOR DESCRIBING A SUPPLEMENTARY BENEFIT OR LIABILITY	B-7
D.1-1	INITIALIZATION (STARTUP) OF ETAM SYSTEM DATA SETS PRIOR TO CONSTRUCTION OF THE ABBREVIATED DATA BASES . .	D.1-4
D.1-2	DETAIL SYSTEM FLOW ABBREVIATED COURSE DATA BASE LOAD . . .	D.1-5
D.1-3	DETAIL SYSTEM FLOW ABBREVIATED VEHICLE DATA BASE LOAD . . .	D.1-7
D.1-4	DETAIL SYSTEM FLOW ABBREVIATED TASK DATA BASE LOAD	D.1-8
D.1-5	DETAIL SYSTEM FLOW PROJECT FILE GENERATION AND RANGE-OF-EFFECT SEARCH	D.1-10
D.1-6	DETAIL SYSTEM FLOW RANGE-OF-EFFECT SEARCH RESULTS PRINT/EDIT	D.1-12

APPENDIX A

STUDY OF DESCRIPTOR AND INDEXING TECHNIQUES FOR ETAM RANGE OF EFFECT ANALYSIS

INTRODUCTION

A major topic in the Educational Technology Assessment Model (ETAM) is a body of knowledge and a set of procedures for determining the range of courses, job-tasks, and instructional vehicles in the Navy that can be affected by a proposed innovation. This range establishes the multipliers of benefits, liabilities and costs that are potential in the innovation, and therefore central to decisions about adoption or rejection.

The initial phase of ETAM specified a taxonomic structure within and between major classes of training entities--courses of instruction, job-tasks, instructional vehicles. Elements in the classification system were each identified and clearly defined. Taken all together, the terms in this classification system were intended to structure the essential functions in training technology. The terms, as sets of functional descriptors, were to be used for the practical indexing of entities such as courses, job-tasks and instructional vehicles in the Navy's inventory. The same set of terms (and their defining concepts) would also be used in "describing" the innovation, or rather in "indexing" the innovation.

The descriptor set used in indexing the innovation could then be applied to the data base inventories of Navy training courses (for example) in making a first pass at selecting those courses that were relevant to the innovation--perhaps 50 out of over 4,000 courses, or 200 out of over 4,000 courses.

This study reports the moving of this background effort towards implementation. A procedure is described for indexing an innovation so as to create the content of a search query to be applied by automatic search on a computerized data base. Techniques are described for indexing data base entities such as training courses, job-tasks and instructional vehicles. The same descriptor set is used that is applied to the innovation. Data about each of the entities are contained in Navy listings with, usually, brief descriptions.

Because the application of subject matter descriptors to innovations and to data base entities with the intent of using one to search on the other requires a good deal of human judgment, the strategies for indexing searches and indexing data base entities are discussed. "Indexing" means the selection of a group of descriptors to apply to the data base entity or to the search specification. Also, a rationale for the search procedures specified for ETAM and for the indexing procedure applied to data base entities is offered. Simplicity for the user is a prime consideration.

The final judgment of relevance or irrelevance of an entity to an innovation is not based on automatic matching of descriptors in the search specification and data base entities. Rather, it is a series of operations in which the human assessor of the value of the innovation examines progressively more contextual information about each candidate entity. The candidate entities are those that survive each next step in winnowing out those that are clearly irrelevant.

Under some specified search conditions, the ETAM level of search is bypassed for direct access to the applicable Navy data base. The identification of entities relevant to the proposed innovation enables specific judgments of magnitude of benefit, liability and cost factors to be taken into account for each entity. There are specific references that can be objectively cited for assessing overall benefits, liabilities and costs. Thus, the validity of the ensuing decision to accept or reject or modify the innovation is given strength; and, if the innovation is adopted, the targets for its application have already been specifically, rather than vaguely, identified.

Recommendations are included that, before widespread adoption, the search and indexing procedures should be tried out and, where necessary, fine-tuned within the defined objectives and theoretical models underlying the present taxonomic structure of ETAM.

PROCEDURES FOR INDEXING THE INNOVATION

It is assumed that the indexer of the innovation or of the data base entities (training courses, instructional vehicles, job-tasks) has training and expertise in the index content and definitions, and has acquired some proficiency in search strategies such as are discussed later in this appendix.

INDEXING THE INNOVATION. The following procedures assume that the indexer is familiar with the definitions of taxonomic terms in the ETAM Phase II-A Final Report, specifically with the procedures described in Task 5 for Determining Range-of-Effect.

The source data which the assessor converts into an indexed description of the innovation may consist of various prose descriptions of the innovation and its intended uses and applications. The assessor may, through his general background in training and training operations in the Navy, amplify or modify this description in his thinking.

The assessor will know that the hits revealed by the indexed search will be only candidates for further examination in determining relevance. The further examination will be made by the assessor looking at the titles of the entities and further contextual descriptions of them.

His indexing policy will tend, therefore, to be generous and lean toward inclusion rather than exclusion of possibilities.

1. Determine Type of Innovation

   Follow the procedures described in the ETAM Phase II-A Final Report, Task 5, in determining the type of innovation--primarily applicable to training courses, job-tasks, or to instructional vehicles. This step may be merely a starting point. It is possible that the course of inquiry may lead to all three classes of entities being explored for potential benefits, liabilities, costs and other effects.

2. Determine Use of ETAM Or Other Data Base

If the innovation applies with certainty only to concretely named entities, bypass ETAM search and go directly to the applicable Navy data base such as CANTRAC on courses, Naval Directory of Training Devices or Occupational Standards on job-tasks. Use of the name of the entity, for example radar ASQ-7, as the key for identifying the relevant training courses or job-tasks or instructional equipment. This is the fastest and most efficient search method--using the most specific and uniquely identifying characteristics of the object search in the search query.

Assume, however, that a subject matter search rather tahn a named entity search is indicated by the characteristics of the innovation. Proceed to Step 3.

3. Describing the Innovation for Search Query

For reasons already stated, the description of the search query may be somewhat broader in scope than would be made by a "tight" characterization of the innovation for purposes other than determining potential range-of-effect.

Select the applicable format type--training courses, job-tasks, or instructional vehicles. (See Figures A-1, A-2 and A-3.) It is advisable to prepare the descriptor selection with a paper format. Put a check mark or circle around a numbered item to indicate selection of a descriptor. If the descriptor is to indicate a NOT characteristic for excluding known non-relevant entities, put a large negative sign in front of the numbered item.

Scan the entire set of descriptors on the format before making selections. The following rules govern the manner in which query descriptors should be formulated.

- o Only an item with a numerical prefix code can be selected. "Categories" of descriptions are underlined (or in capital letters) and merely title a meaningful group or subset of the description.
- o Query descriptors between categories form a logical AND relationship. All must be present in the index descriptor for the entity if selection is to take place. However, if the indexed entity in the ETAM data base does not have an indexed entry under a given category, but the search query does have a selected descriptor under that category, the search operation will ignore that category in selecting or rejecting that entity in the data base. (See examples 1a, 1b and 1c in Figure A-4).
- o Query descriptors within a category form a logical OR relationship. Any one that exists in the search query that is present in the index descriptor for an entity will cause the entity to be selected. (See examples 6a and 6b in Figure A-4.) This assumes, of course, that an AND relationship does not exist due to a query descriptor in another category. (See example 5e in Figure A-4.)

ETAM TRAINING COURSE DESCRIPTORS

Reference knowledge

(01) System purposes
(02) Organizational roles
(03) Contexts of operation
(04) Organizational rules
(05) Other

Enabling knowledge

(10) Operational goal criteria
(11) Nomenclature, identif., location
(12) Procedural descriptions
(13) Job relevant facts, rules
(14) Other

Task formats

(20) Procedure formats
(21) Decision formats
(22) Construction formats
(23) Other

Gross job categories

(30) Operations
(31) Maintenance
(32) Service and administration
(33) Command
(34) Other

Objective task variables as manifest in the training

Equipment and objects used:
(40) Real (41) Simulated (42) Symbolic

Environments in which task is trained:
(43) Real (44) Simulated (45) Symbolic

Tools/instruments used in training:
(46) Real (47) Simulated (48) Symbolic

Reference/enabling information in doing task:
(49) Applied (50) Not Applied

Criteria of task performance:
(51) Real (52) Simulated (53) Symbolic

Task functions dominant in training

(60) Goal projection
(61) Scan-detect
(62) Identify
(63) Interpret
(64) Procedure following
(65) Decide
(66) Construct, plan
(67) Track
(68) Motor performance
(69) Interpersonal interaction
(70) Recall task-cycle information
(71) Recall enabling information
(72) Adapt improvisationally/impromptu
(73) Other

Stage of learning

(80) Orientation, familiarization
(81) Task nomenclature, identifications, locations, facts, rules
(82) Task formats at conceptual level
(83) Procedures at verbal level only
(84) Task components with guidance
(85) Entire job-task procedurally at barely acceptable mastery
(86) Highly proficient in work context
(87) Unusual task conditions

(88) Performance at key man level
(89) Refresher learning

FIGURE A-1. FORMAT OF DESCRIPTORS FOR INDEXING AN INNOVATION OR DATA BASE ENTITIES: TRAINING COURSES

ETAM JOB-TASK DESCRIPTORS

ADMINISTRATIVE

Routine paperwork
(01) Forms filling
(02) Document file-management
(03) Decide-encode
(04) Screen-filter distribute
(05) Other

Non-routine paperwork
(11) Construct messages-reports
(12) Analyze-interpret-evaluate
(13) Construct recommendation-proposal
(14) Construct-plan
(15) Other

Office equipment operation
(21) Typewriter
(22) Telephone, etc.
(23) Reproducer
(24) Computer terminal
(25) Teletype
(26) Other

INTERPERSONAL

Inform-instruct-manage
(31) Brief-debrief
(32) Instruct-train
(33) Assign-monitor-coordinate
(34) Evaluate
(35) Advise-inform-negotiate
(36) Other

TECHNICAL

Procedures
(41) Sequential
(42) Strategic-adaptive
(43) Interpersonal-team member
(44) Other

Type of procedure (main emphasis)
(51) Scan-detect: symbolic (including maps, transduced signals, radar)
(52) Scan-detect: natural events, things
(53) Identify: symbolic (including transduced signals)
(54) Identify: natural events, things

TECHNICAL (Cont'd)

(55) Interpret: symbolic
(56) Interpret: natural
(57) Perceptual-motor
(58) Cognitive operations
(59) Manual
(60) Communicate

With or without equipment
(61) With equipment or applied to equipment (paper is "equipment')
(62) Without equipment

Decide
(71) Diagnose-analyze
(72) Select-choose
(73) Under stress or load
(74) Other

Construct-repair-plan
(81) Manual construct or repair
(82) Cognitive construct-plan

Track-aim-steer
(91) Applicable
(92) Time stress
(93) Information-load stress

FIGURE A-2. FORMAT OF DESCRIPTORS FOR INDEXING AN INNOVATION OR DATA BASE ENTITIES: JOB-TASKS

INSTRUCTIONAL VEHICLES: DESCRIPTOR LIST

Vehicle type

(01) Instructor
(02) Static graphics
(03) Animated graphics
(04) Audio
(05) Physical models, demonstration
(06) Procedural trainers: symbolic
(07) Procedural trainers: physical but not functional
(08) Procedural trainers: functional
(09) Task and system simulators
(10) Real equipment, itself

Class of training objective

(11) Reference knowledge
(12) Knowledge-task specific/enabling
(13) Task-skill formats
(14) Skill training

Vehicle properties

(21) Visual
(22) Auditory
(23) Kinesthetic/vestibular
(24) Tactile

Type of content displayed

(31) Text-verbal
(32) Diagrammatic
(33) Abstracted pictorial representation
(34) Pictorial representations
(35) Physical representations
(36) Other

Type of presentation sequence

(41) Library of frames or items
(42) Presentation sequence not applicable
(43) Fixed sequential frames or items
(44) Random selection of frame sequences
(45) Dynamic change of content within frame

Selection source for sequencing

(51) Internal program
(52) Instructor
(53) Student choice
(54) Student performance
(55) Combinations of the above

Type of external control operated by student

(61) Not applicable directly
(62) Artificial or symbolic response
(63) Representational response by symbolic selection
(64) Reperesentational response by dummy control activation
(65) Task-manipulative response, non-dynamic in time and force
(66) Task-manipulative, dynamic and interactive

Feedback presentation logic

(71) Not applicable
(72) Selects next stimulus item or
(73) Gives evaluation of preceding
(74) Selects and presents guidance information

Response evaluation logic

(81) Not internal--depends on instructor or student evaluation
(82) Evaluation limited to student's immediate response
(83) Evaluation extended to a set of student responses
(84) Tolerance limits on acceptable student responses: FIXED
(85) Tolerance limits on acceptable student responses: VARIABLE

FIGURE A-3. FORMAT OF DESCRIPTORS FOR INDEXING AN INNOVATION OR DATA BASE ENTITIES: INSTRUCTIONAL VEHICLES

EXAMPLE NO.	INNOVATION QUERY DESCRIPTOR	SAMPLE ENTITY INDEX DESCRIPTOR*	SELECTED
1a	03 21	$\underline{03}$ $\underline{21}$ 30	YES
b		$\underline{03}$ $\underline{22}$ 30	NO
c		$\underline{03}$ 30	YES
2a	03	$\underline{03}$ 21 30	YES
b		21 22 30	YES**
3a	21	03 $\underline{21}$ 30	YES
b		$\underline{21}$ 22 30	YES
4a	03 -21	03 $\underline{21}$ 30	NO
b		03 22 30	YES
5a	03 -21 22	$\underline{03}$ $\underline{21}$ 30	NO
b		$\underline{03}$ $\underline{22}$ 30	YES
c		$\underline{03}$ 21 $\underline{22}$ 30	YES
d		$\underline{03}$ 30	YES
e		$\underline{02}$ 21 22	NO
6a	21 22	03 $\underline{21}$ 30	YES
b		03 $\underline{22}$ 30	YES

* Descriptors underlined are primary basis for selection/non-selection.
**No 01-05 index descriptors - therefore entry is selected.

FIGURE A-4. EXAMPLES OF ENTITY SELECTION BASED UPON INNOVATION DESCRIPTORS (TRAINING COURSE DESCRIPTORS)

- A query descriptor will select any entity which does not have any index descriptor in that category, provided any AND relationship is satisfied. (See example 2b in Figure A-4.)

- Select one or more descriptors within a category. It is not necessary to select a descriptor in every category--a search query could conceivably consist of only one descriptor in one category. If a descriptor within a category is selected, it should best fit the innovation. If the assessor is uncertain about which descriptors best fit the innovation and the intent of this stage in search, he may select several descriptors within one category and thereby play it safe. Descriptors selected within the category are logically OR'd to each other. This means that (with respect to this category) the entity will be selected as a hit if either one or the other descriptor (or both) match the index descriptors in the data base entity. This OR relationship is automatically created when the assessor selects more than one descriptor within a single category. He may select as many descriptors within a category as he chooses. The OR relationship among index terms serves to expand the range of entities in a data bases that will be selected.

- The assessor may be aware that if some index descriptor appears in a data base entity, that entity is thereby irrelevant to the search objective. He then enters the number of the descriptor along with a NOT function that is applied to it. This means that if an entity in the data base contains that descriptor in its index, the entity will be rejected no matter how well it matches other search descriptors. Notice that it makes sense to use a NOT descriptor only if no positive descriptors are selected within the same category that contains the NOT descriptor. However, it should be recognized that what may be logically equivalent in the formulation of search arguments may not be psychologically equivalent. In any event, if the assessor inadvertently includes a NOT descriptor in a category which also contains positive valued descriptors, the search logic ignores the NOT item as a redundant and irrelevant specification. (See example 5c in Figure A-4.)

- The descriptor OTHER: Ordinarily in constructing a search specification this descriptor is ignored. But on some occasions, none of the descriptors in a category fits, but the searcher feels strongly that there ought to be a descriptor in this category for the innovation. He may ask for OTHER and be prepared to scan through the miscellany of entities that the indexers of the data base have labeled with this descriptor because they too could not find the right descriptor for it. It is likely to be a long chance. The descriptor OTHER may be used along with other descriptors in the category in order to be sure that all the possibilities have been examined.

- The search specification is readied for entry. The assessor may request--or automatically be presented with--a display or printout of the entire search specification. This display is identical in appearance to the format for selecting descriptors, except that only the selected descriptors--under their category names--are presented. Any NOT descriptors are so identified in the display with a preceding minus sign. The specific search operation is also identified (in human readable symbols) with the innovation's project file number and a search entry code identification. These data and the particular search specification will be retained for retrieval. It should also be carried along with any output listings that result from the data base search.

4. The Search Specification Is Entered

The assessor interactively selects a function that enters the specification into data base search.

5. Preliminary Search Output: Count of Hits

Because the search specification may produce a very large list of hits that would be time-consuming to display, the system begins by presenting an overall count of items that match the search specification. The assessor may recognize at once that something is wrong with his search specification. He may then review and revise it, and resubmit it for search.

6. Inspection of a Sample of Hits

The assessor may request display of a sample of the listing. Perhaps the first twenty-five items are displayed to him. The content of listings in the ETAM data base are limited to the title of the entity and the Navy data base code identification of the entity. In the case of a training course, the CIN/CDP codes and title of the course are presented. In job-tasks, the rating, rate, job-task code number and title according to the Navy's Occupational Standards directory is given. In instructional vehicles, the Directory of Naval Training Devices code identification, Federal Stock Number, and name of the device are presented.

7. Revision of Search Specification

The listing of the previous search results to the assessor may be sufficient for him to revise his search specification. This is likely to make it more constraining. He may resubmit the search.

8. Rejection of Unwanted Entities

In many cases, the brief listing of an entity may provide sufficient context information to enable the assessor to reject the entity from further consideration. An interactive program will be provided to assist the user in the editing of the results of a search. This manual editing will allow deletion or addition of entities as required.

9. Saving the Results of a Search

When the user has made his selection of wanted entities in the results of a search, the results will be automatically saved as a component of the project data base. The results are subject to further perusal or editing as required.

10. Combining the Results of Successive Searches

For the purpose of procedural simplicity, the user is not given the opportunity to formulate complex Boolean search arguments. A complicated pattern of AND and OR relationships requires the formulation of a number of individually simpler search specifications. The user may use the interactive search results editing program to combine search outcomes as required.

11. Getting Further Context from Navy Data Bases

Ordinarily the information gleaned from entity titles is not likely to be sufficient for final decisions of relevance to the innovation. (Ordinarily it requires less information to make the decision of "irrelevance" than the decision of "relevance." The decision alternatives here are "relevant," "irrelevant," and "undecided.") The coding format in ETAM of Navy data base entities may be equivalent to that in the respective Navy data base. If so, a machine readable output from ETAM may serve as direct input to the Navy data base. If not, some manual translations may be required for accessing the computer files. The paper equivalent of the same files may be accessed by the code designators. The latter makes for more nuisance and time in paper shuffling, but is not impossible.

In the final stages of decision about relevance, the assessor will be scanning candidate course descriptions at the level of CANTRAC summaries or more detailed course outlines and schedules, NOTAP (Naval Occupational Task Analysis Program) job-task descriptions lying behind the job-task titles in the Occupational Standards directory, or the descriptions in the Naval Training Devices listings. In the case of instructional vehicles, the TECEP* directory can give still greater assistance in focusing on equipment and media types of relevance to the inquiry.

The fact that the listing of relevant entities will be coded consistent to Navy data base codes for the same entities will enable access to cost, usage and other data essential for the processing of costs and benefits in later ETAM tasks.

---

*Braby, Richard Ed. D.; Henry, James M.; Parrish, William F.; and Swope, William M. Ph.D; 1975. A Technique for Choosing Cost-Effective Delivery Systems, TAEG Report No. 16, Training Analysis and Evaluation Group, Orlando, FL.

12. Entering Results Into the Project Data Base

Search results, edited as required, are saved as a specific file of the project data base for that innovation. The source search arguments that were used as a basis of the search are also preserved as separate files of the same project data base. The contents of either the search arguments or the search results could be applied to other innovations that are equivalent or overlapping in objectives.

COMMENT. This description has not included a variety of user convenience functions at the terminal. These deal with such factors as user ID, file ID, correction of unintended entries, tutorial services dealing both with the procedure itself and with reference to the meanings of each of the descriptors in the descriptor sets. These are important considerations, but outside the present scope of effort to implement.

PROCEDURES FOR INDEXING INSTRUCTIONAL VEHICLES, TRAINING COURSES AND JOB-TASKS

Indexing the subject matter in a data base is a discipline. It requires a working conceptual knowledge of the taxonomic nomenclature, but no nomenclature of subject matter is free of substantial ambiguity. The ETAM indexing nomenclature is no exception.

In ETAM, the objects to be indexed have two orders of ambiguity. Although the intent of ETAM is to index, say, a given instructional vehicle, the indexer must use as his reference not his direct knowledge about that vehicle, but a summary description of the vehicle. The same applies to training courses and to job-tasks. The indexer must be able to interpret the often cryptic and always incomplete descriptions of the objects being referenced. His task, thus, is more difficult than the librarian kind of indexer of documents who can work with the document content at its face value. Assigning a descriptor to an object is therefore an act of interpretive judgment. The judgment centers on the tradeoff between what is likely to be gained for retrieval objectives by adding, or by not adding, a given index term to the title or name of the object.

The ultimate validity of an indexed data base can be tested not by truth but by practicality. Assume the following conditions: A sample of users--in this case, of ETAM assessors--is given unrestricted time in which to search exhaustively through all of the data base content of course titles, course descriptions and other references. The objective of the searches is to match, say, training courses relevant to the knowledge of the specific training innovation. This process results, presumably, in a number of hits on "relevant" objects. Now let equivalent searchers use the subject matter index and the indexed data base for equivalent searches. To the extent that results in hits from the second procedure match those obtained by the first procedure, the index and indexing operations (taken together as they must be) are valid or effective.

A comparison of the time effort and other costs associated with each of the two procedures would be measures of relative efficiency. In the practical world of cost-to-benefit ratios, even substantial reductions from a measure of Utopian validity can be offset by large cost reductions. In the design of search procedures, there tends to be a tradeoff between the average effort spent in putting the needle into the haystack and the average effort that will be spent in finding it, or the cost of not finding it at all.

This argument is intended to motivate the would-be indexer who correctly realizes that the notion of "perfect indexing" is meaningless. In technical jargon, subject matter indexing is in the class of operations that are "heuristic."

INDEXING OBJECTIVES

> Policy 1. The primary objective of indexing the data base is to help the ETAM analyst determine the range-of-effect of a proposed innovation or change to existing (or projected) Navy practices or facilities for training.

The indexer should strive continuously to bear in mind his goal object--to give the ETAM user a practical tool for determining the range-of-effect of a proposed innovation in training technology (whatever its nature). The indexer is creating a job aid. The indexed content is not an end in itself.

A different user and different purposes would certainly generate a different indexing content and structure, and different indexing policies.

> Policy 2. In assigning descriptors, err on the side of including the irrelevant rather than excluding the relevant.

This policy is based on the importance of getting the fullest reasonable range of applicability of the innovation or change on which the data base query is based. Rejects at this search level will not again be candidates for inquiry. The indexer must also remember that his index will generate only a preliminary pass at selecting candidates, and that final decisions to accept or reject objects yielded by the automatic search will be made by the ETAM assessor while examining the full context of course titles, course descriptions and so on.

Obviously this advice must be followed temperately. It applies where the indexer has reason to be ambivalent about including or excluding a given descriptor while indexing an object. He will have to take into account the entire context of the description of the object plus his interpretation of what lies behind the description presented to him. The significance of the descriptor to that context should also weight his decision to accept or reject. If the descriptor, or an approximation of the descriptor, seems to have a central importance to the context (of the course, of the vehicle, of the job-task or job) then he should weight in favor of inclusion and let the assessor later decide on relevance. But if the descriptor seems to apply to a relatively unimportant factor in that particular training, or job or vehicle, there is little practical loss if when he is uncertain he rejects using the descriptor. By so doing, he increases search efficiency for the ultimate user.

Indexing Procedure. The following steps include the preparation by the indexer for indexing the ETAM data base content. It is assumed that the indexer will become a skilled specialist in performing this function as contrasted with a host of persons performing this function only occasionally.

1. Acquire a clear picture of the task objectives of the users of the indexed materials.

   A continuing awareness of the purpose to be served by the indexed content should help the indexer to do what is sufficient and necessary for the ETAM user. The indexer's strategies and decisions can remain clearer and more relevant if purpose is restricted to specific needs. The indexing of content for retrieval is a reciprocal operation to querying and searching. Ideally, the indexer acquires some experience in simulated search problems and thus becomes sensitized to the searcher's difficulties.

2. Learn the nomenclature and concepts of the indexes.

The meaning of each index term is given in the ETAM reference material. In many cases, these meanings are somewhat specific to the ETAM context. For example, the term "decision format" cannot be sensibly used without the specific ETAM definition. The user who queries the data base will use the same reference material for selecting index terms in his queries. It is therefore essential that the person who indexes the data base objects, and the person who indexes queries to that data base, share in a single reference that defines the index terminology. Notice that most index terms are clustered into categories or groups. These groupings have a contextual purpose for both the indexer and the searcher. After a number of hours that combine study and application practice, the indexer will begin to find that the appropriate index terms will often emerge almost automatically when he looks at the reference content that describes the object to be indexed. This automaticity has value in efficient classification; it has a liability in that the indexer may become stereotyped in assigning descriptors when he should be exercising contextual judgment.

3. Learn the search rules and strategies that will be applied to the indexed materials.

The indexing of the data base content is a critical step in determining a range-of-effect. But it is only the first in a series of steps taken by the assessor in seeking to match the description of an innovation with entities identified in the data base and ultimately with real entities--jobs, courses and instructional vehicles. The indexed search operations could result in a collection of "probable" relevant items and the rejection of "improbably" relevant items. The final decision of relevance occurs when the assessor examines the contextual descriptions of the items that have been screened as probably relevant.

The indexer should be intimately familiar with these operations and strategies, and the kinds of judgment the assessor will have to make as he proceeds. The indexer can stipulate that the assessor uses the same definitions and meanings of index terms that he uses and which are contained in the shared glossary reference.

By being aware of the judgmental operations of the assessor and query-maker, the indexer can realistically control criteria for precision in indexing. This awareness can be crucial to the indexer's speed and comfort in assigning descriptor nomenclature to object descriptions. He should, of course, have a realistic knowledge of what happens when a data base search is screened by descriptors in various logical relations--AND, OR and NOT, according to ETAM query construction rules. (See preceding material in this appendix for those rules.)

4. Practice assigning descriptors to several scores of entities in each entity class.

The results of this practice are thrown away. The intent is twofold. One is to master the classificatory format so that the indexer can think with it more or less spontaneously. After a first pass at indexing several dozen entity descriptions (at whatever the level of description is to be used), perform a second and independent pass on the same sample of materials in a day or two--after some degree of forgetting of the first pass. Then compare the results of the first and second passes for reliability. There will be variations, and this would be expected even among highly skilled indexers. But the novice indexer should attempt to think through and account for the differences between his first and his second indexing of the same item. It is not necessary to think of one or the other as being "wrong" when two or more passes on the same material do not match precisely. This thinking through of the variations can lead to a relatively consistent decision policy in assigning descriptors.

5. Index on the basis of entity context as well as specific elements of description.

Assume one is indexing a course title and description. Before assigning any descriptor terms, one should read all of the information that is going to be used (or is available) for the indexing of the course as an entity. One should get the entire context in mind. This enables judgment in selecting index terms that will be useful in a practical way. For example, an operator's course on radar equipment mentions that "routine maintenance operations and diagnosis" is an item of study. The entire course is limited to 10 days. It is a pretty safe bet that the "maintenance diagnostics" will be limited to a few external, procedural tests in order to determine whether the set is operationally usable or not. Almost certainly there will not be circuit and module levels of troubleshooting taught and learned, nor complex inferential strategies applied by the operator. Therefor, the expressions "maintenance" and "diagnosis" in this context would probably not warrant applying to this course, or course segment, nor the descriptors "maintenance" and "decide" (the latter including the meaning of "diagnose") for that course, at least on the basis described here.

The indexer will, of necessity, have to make a number of inferences of this kind. Sometimes his guesses will be wrong. But if he has a general background knowledge of the subject matter (training courses in this example) and uses the contextual information presented to him, his batting average must be better than if he relies only on words and phrases taken out of context.

The descriptor "Other" appears in some categories. This is an escape hatch and should be used with reluctance. On occasion, however, a category may connote concepts that are important for identifying something important about the entity to be indexed, but none of the descriptors in the category make even an approximate fit.

Use the term "Other" in the expectation that a searcher will sometime have a reciprocal difficulty in finding the right descriptor--or several descriptors--and specify "Other." The "Other" file will be a miscellany of items that the searcher will have to look through in terms of their contexts in the hope of finding a hit.

If several indexers agree that the "Other" term is being used too frequently in a given category, the definitions of the other descriptors should be restudied. If this is still unsatisfactory, the definitions of the descriptors may have to be somewhat modified. If this is still unsatisfactory, another descriptor may have to be added to the category.

But changes to the present index, especially after any widespread adoption, should be made with great reluctance. That is because changes will invalidate to some degree the existing indexes of data base entities. A test of motivation to change is willingness to reindex existing entries. This is less a threat if an index term is added to the present list. But as index terms are added, the human difficulties in using the index, either for search or for entities, escalate.

6. Use descriptors to identify key factors in the entity.

Apply a descriptor to an entity (course, job or vehicle) only if the descriptor identifies something important in a practical sense. Some examples will clarify.

Although there can be exceptions to the following principle, it is generally a good one to follow--a factor, feature or element which is less than 5% of the cost, or time or function of the total entity does not justify identification by a descriptor. Thus, a fifteen minute discussion on "system purposes" in a two-week course for operator or maintenance personnel would not justify that the index to these courses include the descriptor "system purposes." But, if a half day or a full day in the two-week course were spent on this topic, it would justify the inclusion of the descriptor.

The 5% principle does not apply if the reference is highly important. The solo pilot may spend less than 5% of total mission time on navigation, but it is a key function to mission effectiveness. This would therefore be an appropriate activity to account for in a job-task characterization--in our indexing system, navigation would be indexed as "interpret" and "cognitive operations."

The data for making these quantitative and qualitative judgments rarely appear in the entity descriptions upon which the indexer generally depends. In consequence, the indexer should have general knowledge and practical acquaintance with the subject matter he is indexing.

Assume that the indexer has general knowledge background of the subject matter being referenced by the description from which he is making the index; assume also that the indexer has thorough familiarity with the meanings of all of the descriptor terms; finally assume that he has had a few dozen hours of experience in applying the descriptor set to these entity descriptions. It should take him between two and five minutes, with a median value of less than three minutes, to index whatever entity is being indexed--the job-task, the course of instruction, or the instructional vehicle. Some entities--such as homogeneous units of instruction--could be indexed far more rapidly--less than a minute. Entities in each of these kinds of materials tend to group themselves so that the indexer learns to recognize what pattern of descriptors applies to members of the subsets. Additional time is required when a standard pattern is not the case.

If the indexer labors for many minutes on a large proportion of individual entities or items, agonizing over the choice of a descriptor, something is wrong. He has not been properly oriented to the task--he may be unduly compulsive about resolving ambiguities, or his information background for the task may be inadequate. On the other hand, if the indexing always proceeds as rapidly as his hand can check off descriptors, he may not be giving enough critical examination and thought to the subject matter. Properly done, indexing is hard mental work. It requires holding a context of information in mind and exercising degrees of judgment not only in what descriptors to select, but also which ones to exclude.

7. Use a standardized format for indexing.

One indexer of samples of job-tasks, courses and instructional vehicles used copies of the format shown in Figures A-1, A-2 and A-3. A special heading was added that specified Navy data base identification code, title of the entity and, as a special factor in course descriptions, course length in days. These data were written by hand and, in many cases, the careful writing (lettering for legibility) often took more time than the indexing of the entity.

A circle drawn around the code number of the descriptor meant the selection of that descriptor. One sheet of paper was used for each entity. It was easy to erase in making corrections or changing a choice. A separate manual step was required for entering these data into the computer.

This indexer felt more comfortable with the entire layout of the descriptor set before him than he would if he had to select or reject descriptors, one at a time, by the prompting method. In fact, this is probably an undesirable procedure. Unless a graphic display can present the entire descriptor pattern all at one time, as well as obviate the need to copy entity titles and codes by hand, the paper format may continue to be preferred.

SEARCH RATIONALES AND STRATEGIES

Consider a data base conceptually like the two parts of a telephone book. On the white pages there appear unique identifiers associated with a given telephone--subscriber name, subscriber address, subscriber telephone number. Assume, like social security numbers, that there are no duplicate subscriber names. Given the subscriber's name, his telephone number can be unequivocally identified. There is no ambiguity in reference. If the directory is up-to-date the telephone number linked to the subscriber's uniquely identifying name or code is evoked without uncertainty.

Think now of using the yellow pages. Assume you wish to find the telephone number of someone who sells mousetraps. There is no subject matter entry called "mousetraps....sellers of--." You must now make inductive and deductive inferences about the class of sellers who sell mousetraps. Perhaps you think of "rodent control" or "department store" or "hardware store." You are making inferences that this class term includes a seller of mousetraps. You may be right in any of these cases, or wrong in some or all of them. Subject matter descriptors are ambiguous even when carefully defined, unless they specifically point out exhaustively all the members of the set in the class name. This latter is an example of a "closed set." Most class names identify open sets. In ETAM indexing we will be talking mainly about class names that apply to open sets of references.

SUBJECT MATTER SEARCH IS HEURISTIC. Subject matter names, as contrasted with unique entity names, are to some degree ambiguous. The assignment of a subject matter term to an entity in a data base requires some judgment, and the selection of a subject matter term for a search query also requires some judgment. Matching a subject matter intent with a subject matter description therefore is not the result of rigorous processes; consequently, success is only probabilistic and may be increased by the use of a different kind of probe. Success probability also increases to the extent that the indexer of the subject matter shares his definition of subject matter terms with the searcher who selects terms for the query.

There may be cases where the searcher's concept of what he is searching for may be appropriately modified either by inspection of the list of permissible index terms (in a closed indexing system) or by examining the content of an early query and consequently changing the terms he has used in previous queries.

That a success is probabilistic in some degree, and may be based on iterative, converging operations, justifies treating subject matter searching as a heuristic rather than as a rigorous operation. This fact should make a difference in how the search procedure, and its information support, is designed.

SUBJECT MATTER SEARCH IS STRATEGIC. This means that some policies in procedure will be more efficient or more effective, or both, than other policies. Strategies are formulated on the basis of tradeoff variables which may be optimized formally (as by equations) or informally (as by human judgment).

The purpose of search is to find in some universe of entities (such as all Navy jobtask in a data base of job-tasks) only those entries that fit the search objective and all of those entries that fit the search objective. The search objective is not necessarily the equivalent of a search specification. The search objective is what the searcher wants or needs to find in the context of a specific innovation; the search specification is his translation of this need or requirement into descriptors and logical connectors among search descriptors that come from the index lists.

We must assume that the searcher can recognize an entity appropriate to his objective if he can inspect all of the information context that is associated with that entity such as, in this case, a full description of the task--its setting, tools and equipment and objects worked on or with, the work functions performed, environment performed in, and explanatory text.

Assume that the data base has 30,000 job-task titles (entities) and associated with each job-task title is a set of descriptors, an abstract, and a full textual description of the job-task entity. An inefficient, but highly effective way of searching, would be reading each and all of the full text descriptions for all 30,000 job-tasks. The next more efficient method, but less certain of complete effectiveness, would consist of reading all abstracts and, only if the abstract sounded promising, to read the full text description about the entity. This procedure entails some risk of missing desired entities as well as selecting entities that turn out to be irrelevant, because the abstract is an incomplete context and susceptible to misrepresentation.

The next more efficient method, but correspondingly more risky, is to make the first winnowing on the basis of index terms assigned to entities--descriptors. (It is possible that an intermediate method would consist of examining merely the titles of the 30,000 entities by a human interpreter of the probable meaning of each task title.)

We can see that there is a tradeoff between the degree of search efficiency--the labor spent in the search--and the degree of search effectiveness (getting all of the relevant entities in the universe of entities). This tradeoff extends to the size of the index list and the complexity of descriptor relationships that are assigned to entities in the data base.

There is another kind of tradeoff--this is between the precision and extent of effort in indexing the entities, and the relative simplicity and effectiveness of search. The greater the expertise and effort spent in indexing the data base content, the less the effort (other things equal) in search, assuming equivalent criteria for search effectiveness.

In the search process there is a tradeoff between making a search mesh so broad that there will be many false drops (items than turn out to be irrelevant) but does capture a large proportion of all relevant entities, as contrasted with a search mesh that is too fine and, although producing few false drops, has also filtered out a substantial proportion of relative items.

It should be evident that the design of an indexing and search technique is a bundle of compromises that hopefully are based on enlightened practicality.

In principle, search strategy produces a progressive winnowing cut of the irrelevant. It is a convergence towards the desired subset through a process of exclusion. It is much like the skilled playing of the game called Twenty Questions where you find what you want by progressively excluding, logically, what you do not want. Unfortunately, this assertion cannot be taken too literally else we are back at reading every document in the library.

SUBJECT MATTER DESCRIPTORS. The following considerations were deliberately taken into account in developing the descriptor list. The descriptor list must serve both the indexer of the content of the data base and the searcher who will query the data base.

Closed Set of Descriptors for a Topic Domain. Recall that topic domain consist of courses and course characteristics--instructional vehicles and their characteristics, and job-tasks and their characteristics. Each domain has about 50 descriptors. A closed and relatively small set simplifies the computer operations of setting up records and fields and search algorithms that are both simple and efficient. All terms can be (and are) carefully defined for both indexer and searcher, thus their usage is relatively standardized. The potential loss of specificity and flexibility in entity description are offset by the ease with which the meanings of the descriptors in this limited set can be learned and their meanings understood and remembered, and their application standardized. Application of the present set of descriptors may show empirically the desirability of discarding, modifying or supplementing members in the list. These changes can be made while still retaining a closed set. By definition, a closed set is under control. We should remember that the topical subject matter level of search is not generally intended to be sufficient in itself to identify or reject target hits as relevant to the search objective.

Few Descriptors per Category. With some exceptions, a category of description contains fewer than eight descriptors. Several sets go to ten or a dozen. A set of this small size is readily learned and its members discriminated from each other. Because more than one descriptor in a category may be used in indexing a reference entity or in creating a query, there is little need to avoid redundancy or seek mutual exclusivity in the descriptor set. By avoiding such a requirement, indexing and querying are simplified perhaps by an order of magnitude. This simplification offsets the loss of precision for a given search iteration. Topical exhaustiveness for a collection of subject matter descriptors is perhaps more important than precision in this iteration of search. But even the exhaustiveness criterion is a hope that, in most searches, the category "Other" will get few hits. If not (and the searcher has AND'd the descriptor "Other" with a number of additional descriptors, thus limiting the target area), the searcher may have to use a different strategy.

If the descriptor "Other" in a set of descriptors is used too frequently, a revision of the descriptor list is indicated. A benchmark for "frequent" might be 10% of queries in which "Other" is the only descriptor selected in a category of descriptors.

Few Descriptors in an Entire Topical Domain. Fifty descriptors, subset into at least a half-dozen topical groups, are fairly readily learned, especially if they make sense as individual descriptors and in the way they are organized. The entire fifty fit rather readily on a single printed page or display, and can be shown all at once to the indexer or the searcher. Thus, the data entry format for composing a query may begin as a printed page with blanks, or as a display on a graphic terminal.

Avoidance of Descriptor Hierarchies. When they are used to compose a query, the descriptors are all at one logical level. It is well known that attempts to make and use formal hierarchical relationships among functional or topical concepts create procedural difficulties for the user, and put some serious constraints on computer search for at least some kinds of search content. In any event, the small number of descriptors per search domain makes hierarchic structure unnecessary either for the human (except as noted) and for the computer system. All of the descriptors can be treated as a single level list.

TAXONOMIC REFERENCE OF DESCRIPTORS. The source of the present sets of descriptors was not merely convenience for indexing. The source is an applied theory of learning--the essential processes, products and supports for the learning of useful tasks. Each term in each set has a somewhat different meaning for a learning process or outcome than other terms in the set. Each term also derives some of its meaning from its contextual association with the other terms in the clusters and sets. The indexer's task will be somewhat simplified if he recognizes the structure inherent in each set, rather than treating them as an arbitrary list created merely for indexing convenience. Their primary purpose, individually and collectively, is to apply standard names for concepts that generalize an instance of a set of properties, such as found in the innovation, to a some universe of entities that share those properties in a practical as well as theoretical sense.

For a more complete exposition of these ideas, see Task 5, Section III in the ETAM Phase II-A Final Report, as well as Appendix A to that report.

Any standardized terminology is bound to have examples with which it cannot readily cope. If the exceptions are rare--perhaps less than 5%--the standards are likely to be useful. The penalty is some proportion of extra human judgment for coping with these exceptions and perhaps putting them into a "miscellenaous" file. If, however, the exceptions approach the frequency of items that readily fit the standards, something is wrong with the standards or with the way in which they are applied.

These comments are intended to apply to some who may feel compulsively driven to find a pre-established niche for 100% of the cases in some real world universe. There will be temptations to add terms to the descriptor set in order to fit perhaps small subsets of entities. The value of a closed set of descriptors can be lost if the set increases in size beyond those that can be kept readily in mind after a few hours of practice.

If changes or additions are contemplated--and indeed they should be from time-to-time--they should be reviewed in terms of benefits and liabilities to the entire logic of the classification structure. It is easy to anticipate many sources of pressure to make changes. Very possibly the entire taxonomic logic underlying the descriptor set justifies technical challenge. It would be far better from economic and other reasons to make sufficient tryouts so that the entire descriptor set could be strongly stabilized at the outset before general adoption in practice.

SEARCH ALTERNATIVES. The searcher will be advised to examine his search objectives and the information about the innovation he has as starting point, and choose his search method accordingly. He should be able to bypass any operations that interfere with his progress by demanding that he go to a higher level of description than he already has available for bounding his search region. (Search strategy will be discussed in a later topic.) In any event, the searcher should not be required to enter any level of inquiry that obfuscates his available definition of his search problem. Furthermore, search procedures should not demand his going through any process that does not make a contribution either in range or scope of potential relevance, or in narrowing the search universe towards his objective. He should be able to bypass non-contributory steps in search procedure. The ETAM search procedures enable this flexibility.

Thus, the searcher who wants to get the broadest topical definition of his inquiry should be guided by the ETAM search descriptors in doing so. The ETAM search will yield a broad set of candidate job-tasks, course titles, instructional devices. This first set of candidates, if reasonable in number, may be examined individually at the level of job-task names, or course titles, or titles of instructional vehicles. On the other hand, the searcher may wish to--and be able to--use descriptors in the respective Navy data bases for further subsetting the first list of candidates into a second derived list of still more probably relevant candidates. He may therefore use descriptor names appropriate to automatic search in those data bases, e.g., the Navy's Occupational Standards data base, NOTAP, CANTRAC, etc., permit, in varying degrees, this kind of search.

The searcher may elect to go directly to the Navy data bases for inquiry based on open descriptor sets applied to these respective data bases. Entity titles and terms within titles can be used to construct search arguments. Thus, many task titles in NOTAP include the name of the equipment, tool or instrument on which or with which the task is performed. After the searcher makes a culling pass through the titles of entities, printed out or displayed as a consequence of the query, the searcher may make another pass through the remaining candidates with the supplemental context for each entity presented to him. This would be the final search pass for accepting or rejecting candidates that had survived the progression of excluding filters imposed by successive search queries.

If the automatic search system is interactive and on-line, a number of successive search passes can be made in perhaps a few minutes, so that the process is neither as complicated nor as lengthy as the description of it.

It should be clear that the searcher is not forced to take any one search route from start to finish. It should be noted that this kind of user flexibility is somewhat unusual in query systems.

SEARCH PROCEDURES. The major factors that make formulating a query easy or difficult are selecting descriptors, selecting logical connectives between descriptors, and composing the content of the query for entry into the computer. The descriptor issue has already been discussed. Because the total descriptor lists are so relatively small for each domain there is no justification for introducing the complexities of hierarchic structure. What is called the "coordinate descriptor" method is being used for creating a query. The user merely enters those descriptors that have relevance to the identification or positive rejection of candidate entities in the data base. If a descriptor is not selected, it is a "don't care" variable in the computer's search logic. Descriptors are organized into meaningful clusters called "categories." The cluster title communicates the context meaning to the user, thus simplifying his interpretation of the meaning of the descriptors within the cluster. In itself, the category title is merely a part of the format display; it does not enter into the search argument unless it is also a numbered descriptor, because there are no descriptors logically subordinate to it in the descriptor set.

The query format itself allows the procedural equivalent of a check list operation in selecting descriptors. This format is desirable whether the user is a novice or expert to the system. Because it is a standardized presentation to the searcher, the descriptor content can rather quickly be learned by the user in repeated usage.

Research (some of it conducted in IBM) has shown that, at least in programming, the use of logical connectives such as AND, AND NOT, OR and EXCLUSIVE OR, is fraught with human error, even among experts. The logical OR, especially in combination with bracketed expressions, is particularly inviting to human misunderstanding and error. These experiences have been taken seriously in formulating what orders of connective logic are permitted the user, and not offered to the user, in making up a single query.

The AND logic is ordinarily simple for the user to get accustomed to, especially if no compound statements are to be strung together. (The procedures do not allow compound statements.) Thus, the searcher may want to get a listing of all courses that provide Enabling Knowledge of "Procedural descriptions" AND in the Gross Job category of "Maintenance." The user quickly learns that only those items which are identified both as teaching procedural descriptions and that teach in the context of maintenance jobs will be selected by the query. (In preliminary training of the user, it would not be taken for granted that even this concept is self-evident.) The AND operation does not require terms in an expression to be logically grouped and sequentially ordered in any fashion.

The OR relationship has been arbitrarily allowed to apply to single descriptors, taken one at a time, within the same category. Descriptors within a category cannot logically be grouped by AND connectors and OR'd to another group of logically connected descriptors. The OR relationship can apply only between individual descriptors within a category. The selected content of one category cannot be OR'd with the content of another category. This would raise the potential logical complexity in a single query beyond what is felt to be a generally acceptable level.

The OR relationship within a category can be useful for several purposes. Assume that the searcher is not quite sure whether the task function he is concerned with is "Interpret" or "Decide." That is, he is not quite sure which of these terms was used to index the target material for which he is searching. To be safe, he can select both Interpret and Decide, which means that he will get entities that bear either one or the other term, or both. He can thus defer his decision on precise relevance until he sees further contextual information about the candidate items selected by his query. Note that this capability copes with some degree of ambiguity in descriptors, both as applied by indexers and by searchers.

The distinction between Inclusive OR and Exclusive OR was not considered operationally relevant to the ETAM application. The OR in ETAM procedures is an Inclusive OR--it does not care whether either one or the other, or both, the OR'd items describe the target entity.

If the searcher wants to widen his search universe more broadly than is permitted by the logical limitations in a single query format, he may use several queries, each of which is the equivalent to a complex OR to the other queries, and combine the results of the several queries. This avoids the convoluted thinking that is required for making a single, multilevel, search statement. If the cost of making even occasional logical errors is prorated across the additional time for making multiple queries, it is doubtful that any efficiency advantage would appear for the more complex procedures. "Logical elegance" in statement is not a relevant human factors criterion.

Finally, some comments are deserved about the ability to frame any query into the content of a single displayed page that is uncluttered with computerese jargon, and symbols that have no bearing on the user's objective--to formulate a search statement on the variables in which he is interested. It should be simple for the user to check over what he has entered, to know what he has done, and if he can recognize errors of omission or commission to correct them directly and at once; and if the response to the searcher's query appears to him like a tray full of nonsense, he can readily check back over the total content of the query he entered as a search argument and perhaps identify what was wrong with it. Thus, the data entry format and its contents is a useful documentation of the query both for diagnosing limitations in the search argument, or for purposefully modifying it.

The test of these rationales will come, of course, when samples of novice users are actually confronted with formats, terminals, and search problems. Assume familiarity with the meaning of the descriptors and with the purpose of the query operations. Within 15 to 60 minutes, the novice should be entering meaningful queries without assistance, and on the average make fewer than perhaps 5% errors in procedure that are not identified and corrected at once.

SEARCH STRATEGIES. The design of query structures, search systems and data bases should facilitate flexibility in the searcher's method of attack. That method may vary depending on the values in the search objective and on the information the searcher has at the beginning of the search. Both of these issues are joined in principles that embody search strategy. The definition of strategy here is to "optimize" an outcome for some given level of risk and effort, or conversely to minimize risk and effort in achieving some level of search outcome. These concepts should become clarified when some examples are examined of efficient search technique under various kinds of starting information.

The general strategic principle is to formulate queries at each stage of inquiry that eliminate the largest proportions of irrelevant entities from those entities that still remain to be searched. In practice, this means using descriptors that appear most rarely (as compared with other descriptors) in data base descriptions. In general, a unique identifier such as the "name" of a given entity is comparatively rare as an index term in a data base. An example in a job-task inventory would be the name of a piece of operational equipment such as "radar ABC-10." Note also that unlike entity attributes where meanings are inevitably ambiguous, a unique identifier is low in ambiguity. If the searcher knows that all of his ultimate target set of relevant entities will include "radar ABC-10" in their indexed descriptions, his first search query should include "radar ABC-10" as a search term. In this case, or one like it, the searcher would bypass the ETAM level of subject matter search and put his inquiry directly into the job-task inventory data base. To repeat, this assumes that entities in the job-task inventory are indexed by "equipment worked on" and that "radar ABC-10" is the descriptor, and the only descriptor, for that equipment, applied to relevant jobtask entities in that data base. (It is assumed that if a descriptor is assigned to an entity in a data base, that automatic search can be made on that descriptor or category of descriptors.) Call a descriptor that is a specific name for an entity, or for a set of equivalent entities, by the term "nominal descriptor." A query can include a number of nominal entities that are logically "OR'd" to each other if the query format permits, else a sequence of queries may achieve the same end.

The use of nominal descriptors does not entirely rule out the missing of relevant items. It is frequently possible that in the operational setting, a given object is treated as a component in a larger object, and only the latter is named as the descriptor. Thus, "radar ABC-10" may sometimes be a component in "radar system XYZ," and, if so, only the latter is identified. This is why the rules used by the indexer should be explicit and communicated to the searcher. Under conditions of uncertainty, and where is is important to determine maximum range of relevant entities, the only recourse is by proceeding from the more inclusive categories into contextual search. Contextual search is made when the searcher manually examines the full description of each entity after automatic search on descriptors has been carried as far as it is allowed safely to go.

When on the other hand the searcher is not sure of what he wants, or lacks nominal descriptors for narrowing the search quickly, he is advised to go broad at the start. This is because if relevant items are winnowed out at any stage of progressive search, they will not be examined again during that particular search cycle. Again, the tradeoff is between the risk of missing relevant items versus amount of effort spent in the total search and identify operation. The selection of an improperly limiting attribute in an early query can lock out a large proportion of eligible entities. In bibliographic search the searcher may not care about seeing the "complete" set of potentially relevant documents, only some subset which is representative and falling within the limits of what he can assimilate. But in range-of-effect analysis, the searcher is properly concerned with the fullest possible range of applicability of the innovation. He is concerned with the application "universe," not a sample from it.

But even with this objective the maximum exclusion principle operates. When the searcher adds descriptors linked by AND, he is superimposing filters on the search. A carefully chosen AND NOT descriptor can filter out large subsets of data base entities that are irrelevant to the inquiry.

An ideal inquiry system would enable the searcher to examine an output listing (whatever its level or form) and randomly sample from its contents at the level of contextual detail. By this means he could often determine what further restrictions to put into his next queries. Note that here again there is a difference betwen bibliographic searches and range-of-effect search. In the former, the searcher may ask for the number of titles identified by the request applied to the data base. If the number exceeds some arbitrary limit set by the searcher, he may tighten his inquiry in an attempt to restrict the number of hits to a number he feels able to manage. The ETAM assessor, on the other hand, is obliged to attend to all relevant entities.

Even when the searcher is quite confident at the start that he can subset the data base with a definitive set of descriptors, he may be well advised to begin with a broader net. The starting assumptions of the relevant characteristics and boundaries of target entities may be revealed as too narrow. Inspecting even a few samples of context about entities not thought to be applicable could change preconceptions and enlarge the field. This might be called "strategic serendipity." A handy--and pertinent--example is "diagnostic technique" or troubleshooting. An innovation may, at the start, be identified as a training method for strategic diagnosis of equipment ailments. The method may be equally applicable to the diagnosis of human ailments. It is possible that some context in one or both may reveal that "search strategy" of the kind we are discussing in the present topic is equally applicable when appropriately defined--the strategic selection of a series of tests (queries) where each test in the series logically excludes from further suspicion (or relevance for further inquiry) some major segment of the equipment (data base). The potential applicability, hence value, of the innovation becomes greatly expanded.

In conclusion, search for information can combine efficiency and effectiveness, but depend on a combination of expertise and enabling knowledge. The knowledge applies to the defining concepts of the descriptor set, the rules used for indexing the target subject matter, and something about the contexts of those subject matters. The expertise applies to strategies used in searching large data banks as well as in overcoming information deficiencies that may exist in the searcher himself. As in other tasks that use strategies, the expert may achieve several times the results of a novice at a fraction of the time.

GENERAL CONCLUSIONS. These rationales are offered as justification for the decisions made in the section and formatting of ETAM descriptors and ETAM query procedures where simplicity for the user was traded off against semantic precision and logical universality. The user is offered flexibility in search method by not being ruled from proceeding directly to data base content if the knowledge of target objectives enables this to be done. The basis for choosing an alternative was described in the context of search strategies.

The functions of the human component in the search operation have been made explicit so that even in the default of automatic modes for aiding search, the searcher could do the job manually with file cards. In any event, the final judgment of the relevance or irrelevance of target entities must depend on human judgment operating on contextual information about those entities. That context may be in a printout or in the searcher's head, or a combination of both.

The combinations of all of these factors makes up an information search "system." The structure proposed here is sufficiently modular and simple in its interfaces and content to be readily changed when actual usage suggests modifications. It is realistic to expect demand for changes, but it is important to determine whether the change should be in elements of method, information content, or the rationales behind them.

APPENDIX B

KEY FIGURES AND ILLUSTRATIONS REFERENCED IN THIS ETAM REPORT

FIGURE	TITLE	REFERENCE *
B-1	Summarized Preliminary Feasibility Profile	III-48
B-2	Cost/Savings Data Sheet	III-56
B-3	Decision Tree for Assessment of Training Innovation (Example of Form Completed with Initial Outcome Values and Probabilities)	III-60
B-4	Decision Tree for Assessment of Training Innovation (Example of Folding Back Process)	III-61
B-5	Format for Describing a Supplementary Benefit or Liability	III-138

---

* The pages referenced in this Appendix are from:
Miller, Robert B. and Duffy, Larry R., 1975. Design of Training Systems Phase II-A Final Report. TAEG Report No. 12-3, Training Analysis and Evaluation Group, Orlando, FL. Chapter III.

SUBTASK	RISK DEGREE / RISK ITEM	UNACCEPTABLY HIGH RISK OR OF TRIVIAL IMPORTANCE	MODERATE RISK OR OF MODERATE IMPORTANCE	NO RISK OR OF SUBSTANTIAL IMPORTANCE	SUPPORTIVE OR OF HIGH OR CRUCIAL IMPORTANCE	COMMENTS, NATURE OF RISK, ETC.	NUMBER OF RISK REDUCTION PROJECTS REQ'D AND IDENTIFIED	ESTMATED PROJECT COSTS
01	IMPORTANCE TO PROJECTED NAVY MISSION							
02	ORGANIZATIONAL COMPATIBILITY							
03	GOALS/POLICY COMPATIBILITY							
04	STATE-OF-THE-ART							
05	R&D FUNDING REQUIREMENTS							
06	TECHNICAL SUPPORT REQUIREMENT							
07	ATTITUDINAL ACCEPTANCE							
	OVERALL PROFILE AND TOTAL PROJECTS AND DOLLARS							

FIGURE B-1. SUMMARIZED PRELIMINARY FEASIBILITY PROFILE

PRELIMINARY
COST ESTIMATE

NEW ☐ **REVISED** ☐

PROJECT	PROJECT NAME	PROJECT NO.	DATE

PERSONNEL				
JOB	MAN MONTHS	COST/MAN MONTH	JOB COST	
				$
EQUIPMENT				
DESCRIPTION	QUANTITY	UNIT COST	COST	
				$
SPACE				
REASON	SQUARE FEET	COST/SQ. FT.	COST	
				$
TRAVEL (NON-STUDENT)				
FROM - TO - REASON	NO. OF TRIPS	COST/TRIP	COST	
				$
STUDENTS				
NO. OF STUDENTS - REASON	NO. OF DAYS	COST/STUDENT DAY	COST	
				$
OTHER				
DESCRIPTION	MAN MONTHS/QTY.	UNIT COST	COST	
				$
			TOTAL PROJECT COSTS	$

FIGURE B–2. COST/SAVINGS DATA SHEET

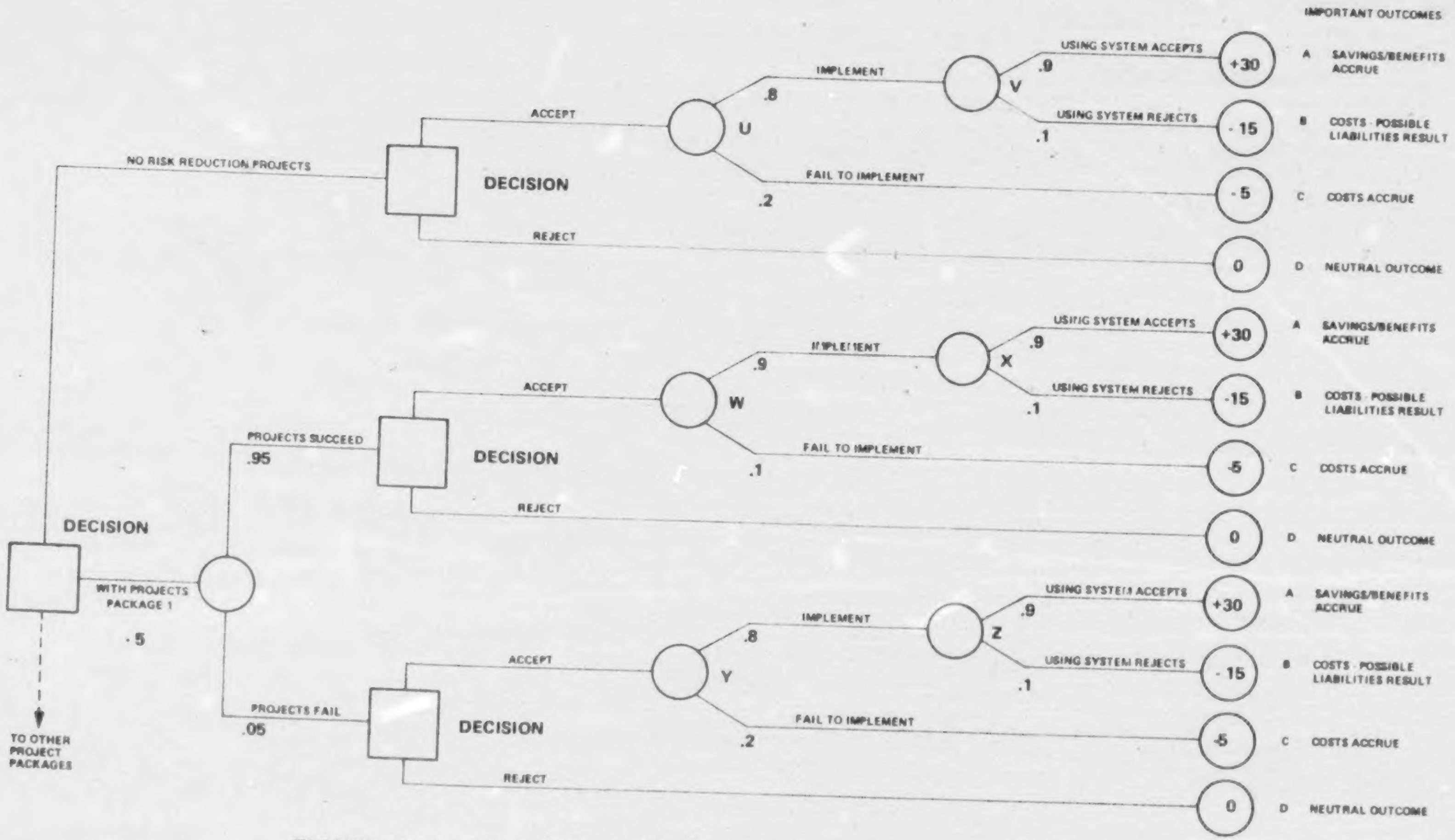

IMPORTANT OUTCOMES
NO RISK REDUCTION PROJECTS
DECISION
ACCEPT
REJECT
U
.8
.2
IMPLEMENT
FAIL TO IMPLEMENT
V
.9
.1
USING SYSTEM ACCEPTS
USING SYSTEM REJECTS
+30
- 15
- 5
0
A SAVINGS/BENEFITS ACCRUE
B COSTS - POSSIBLE LIABILITIES RESULT
C COSTS ACCRUE
D NEUTRAL OUTCOME
PROJECTS SUCCEED
.95
DECISION
ACCEPT
REJECT
W
.9
.1
IMPLEMENT
FAIL TO IMPLEMENT
X
.9
.1
USING SYSTEM ACCEPTS
USING SYSTEM REJECTS
+30
-15
-5
0
A SAVINGS/BENEFITS ACCRUE
B COSTS - POSSIBLE LIABILITIES RESULT
C COSTS ACCRUE
D NEUTRAL OUTCOME
DECISION
WITH PROJECTS PACKAGE 1
.5
TO OTHER PROJECT PACKAGES
PROJECTS FAIL
.05
DECISION
ACCEPT
REJECT
Y
.8
.2
IMPLEMENT
FAIL TO IMPLEMENT
Z
.9
.1
USING SYSTEM ACCEPTS
USING SYSTEM REJECTS
+30
- 15
-5
0
A SAVINGS/BENEFITS ACCRUE
B COSTS - POSSIBLE LIABILITIES RESULT
C COSTS ACCRUE
D NEUTRAL OUTCOME

FIGURE B–3. DECISION TREE FOR ASSESSMENT OF TRAINING INNOVATION (EXAMPLE OF FORM COMPLETED WITH INITIAL OUTCOME VALUES AND PROBABILITIES)

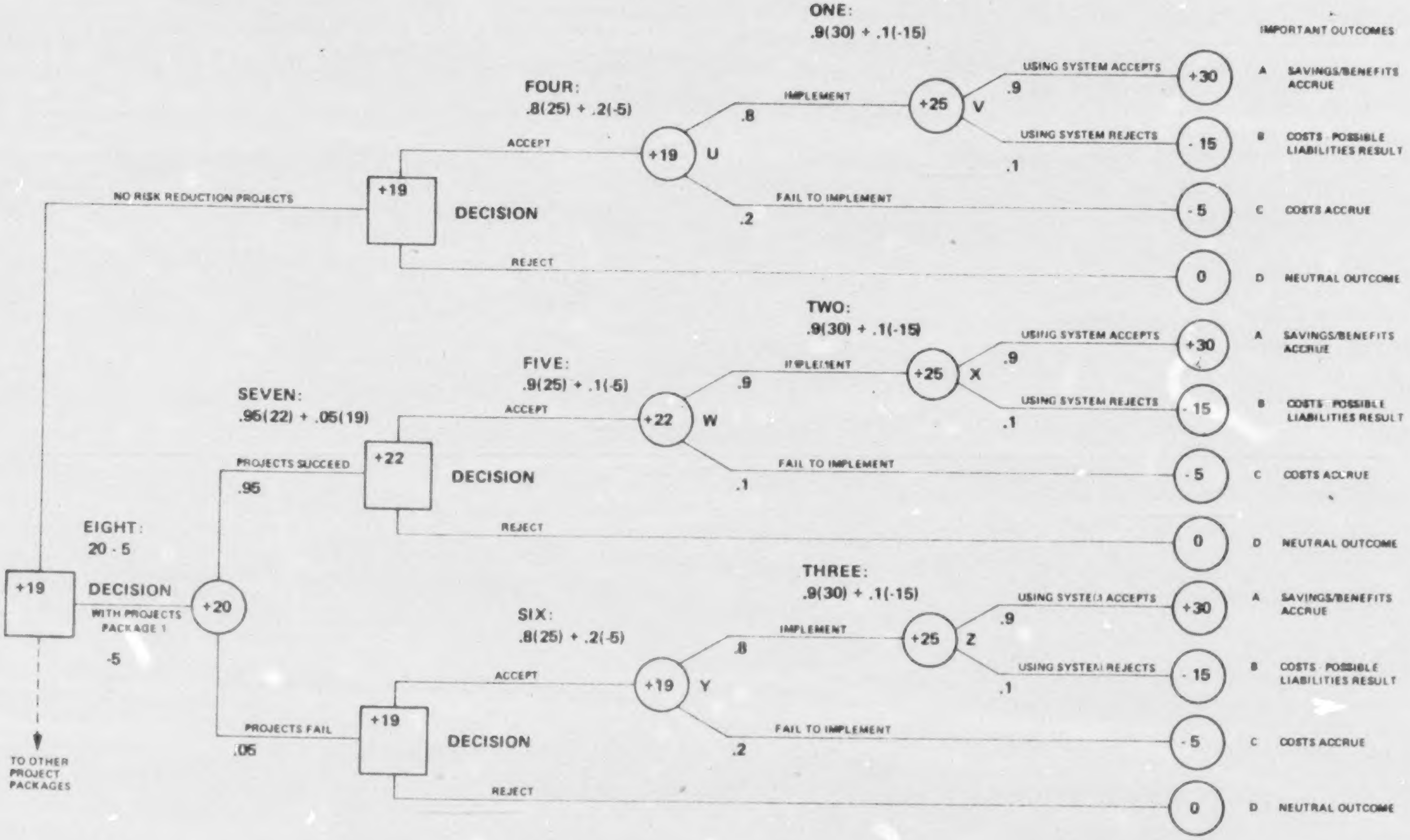

ONE:
.9(30) + .1(-15)
IMPORTANT OUTCOMES
FOUR:
.8(25) + .2(-5)
IMPLEMENT
+25
V
.9
USING SYSTEM ACCEPTS
+30
A SAVINGS/BENEFITS ACCRUE
.8
ACCEPT
+19
U
USING SYSTEM REJECTS
-15
B COSTS · POSSIBLE LIABILITIES RESULT
.1
NO RISK REDUCTION PROJECTS
+19
DECISION
FAIL TO IMPLEMENT
.2
-5
C COSTS ACCRUE
REJECT
0
D NEUTRAL OUTCOME
TWO:
.9(30) + .1(-15)
FIVE:
.9(25) + .1(-5)
IMPLEMENT
+25
X
.9
USING SYSTEM ACCEPTS
+30
A SAVINGS/BENEFITS ACCRUE
SEVEN:
.95(22) + .05(19)
.9
ACCEPT
+22
W
USING SYSTEM REJECTS
-15
B COSTS · POSSIBLE LIABILITIES RESULT
.1
PROJECTS SUCCEED
.95
+22
DECISION
FAIL TO IMPLEMENT
.1
-5
C COSTS ACCRUE
REJECT
0
D NEUTRAL OUTCOME
EIGHT:
20 - 5
+19
DECISION
WITH PROJECTS PACKAGE 1
-5
+20
THREE:
.9(30) + .1(-15)
SIX:
.8(25) + .2(-5)
IMPLEMENT
+25
Z
.9
USING SYSTEM ACCEPTS
+30
A SAVINGS/BENEFITS ACCRUE
.8
ACCEPT
+19
Y
USING SYSTEM REJECTS
-15
B COSTS · POSSIBLE LIABILITIES RESULT
.1
PROJECTS FAIL
.05
+19
DECISION
FAIL TO IMPLEMENT
.2
-5
C COSTS ACCRUE
REJECT
0
D NEUTRAL OUTCOME
TO OTHER PROJECT PACKAGES

FIGURE B–4. DECISION TREE FOR ASSESSMENT OF TRAINING INNOVATION (EXAMPLE OF FOLDING BACK PROCESS)

TASK: DETERMINE RANGE-OF-EFFECT

1. Identify the name of the variable or factor which is a supplemental benefit or a liability.

2. Estimate the magnitude of effect of the benefit/liability.

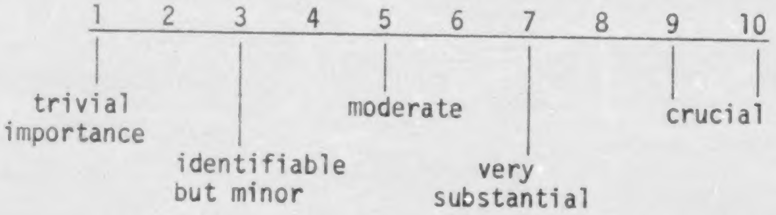
   a. If possible, quantify the unit of magnitude, and the multiplier factor, to apply to the unit of magnitude appropriate to the intended scope of the application of the innovation.

   b. If numerical quantification is impractical, express magnitudes in relative terms.

3. What is the number of entities or instances (jobs, courses, students, etc.) to which this magnitude of benefit (or liability) applies?

4. Estimate the relative importance of the added benefit (or liability) to the overall effectiveness of the innovation.

   Use a scale from 1 to 10.



   Any additional statements that may add or qualify information for the decision maker may be added informally.

5. Estimate the range across which the magnitude of benefit (or liability) may occur.

   a. The estimate of magnitude made in item 2 presumably assumed conditions moderately favorable to the benefit occurring (or moderately unfavorable for the liability to occur) at that magnitude.

   b. Estimate the magnitude of the benefit under conditions highly favorable for the innovation. (The converse for a liability.)

   c. Estimate the magnitude of the benefit under conditions highly adverse to the innovation.

6. State reasons and justifications for these expectations.

FIGURE B–5. FORMAT FOR DESCRIBING A SUPPLEMENTARY BENEFIT OR LIABILITY

APPENDIX C

TRAINING COST MODEL PROGRAM

```
C     **************************************************************************
C     *                 PROGRAM - TRAINING COST MODEL                          *
C     *                 TECEP COST MODEL ADAPTATIONS FOR ETAM BY L. DUFFY      *
C     *                                                                        *
C     *                 THIS MODULE MUST BE MODIFIED WITH NEW READ AND         *
C     *                 FORMAT STATEMENTS TO RECEIVE DATA FROM THE             *
C     *                 ORIGINAL AND MODIFIED EXC FILE FROM THE ETAM           *
C     *                 PROJECT DATA BASE.                                     *
C     *                 THE MODULE PRESENTLY CYCLES AND DEVELOPS A             *
C     *                 MATRIX OF INVESTMENT AND RECURRING COST/SAVINGS        *
C     *                 DATA FOR EACH COURSE RUN UNDER BASELINE CONDITIONS     *
C     *                 AND THEN WITH THE BENEFIT PATTERN APPLIED.             *
C     *                                                                        *
C     *                 DATE - 3/4/77                                          *
C     *                                                                        *
C     **************************************************************************
C
C     PROGRAM SETUP
C
      REAL INTSPO,IMODEV,MSCSPO,MSCSTD,IMDMNT,INSTSL,INFLAT,INRATE
      REAL NSTUD,NTRAV,NINSTR,NADMIN,NFACIL,NSUPPL,NMISC,NEQUIP,NIMD
      REAL ISTUD,ITRAV,IINSTR,IADMIN,IFACIL,ISUPPL,IMISC,IEQUIP,IIMD
      INTEGER GRAD,PURCHF,AEQUIP,DEPF
      INTEGER DIGIT(20) /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,
     1                   17,18,19,20/
      LOGICAL RPT1,RPT2,RPT3,RPT4,RPT5,RPT6,RPT7,LAST
      DIMENSION STUD(20),GRAD(20),STUDMW(20),SMWRRC(20),AOB(20),PSP(20)
      DIMENSION RINSTR(20),RADMIN(20),REQUIP(20),RFACIL(20),IMDMNT(20)
      DIMENSION RSUPPL(20),RMISC(20),CSTUD(20),CTRAV(20),CINSTR(20)
      DIMENSION CADMIN(20),CEQUIP(20),PEQUIP(20),PDEP(20),AEQUIP(20)
      DIMENSION COPMT(20),CFACIL(20),CIMDMT(20),CSUPPL(20)
      DIMENSION CMISC(20),DISCNT(20),INFLAT(20),CGRAD(20),CSTIN(20)
      DIMENSION CSP(20),MATA(100),MATB(100),COPTOT(20),MGRAD(20)
      DIMENSION AOBX(20)
      DIMENSION ETAINV(20),ETAREC(20),ETAGRD(20),ETASTU(20),ETAAOB(20)
      DIMENSION ETAINS(20),ETAADM(20),ETAEQP(20),ETAFAC(20),ETASUP(20)
      DIMENSION ETAMIS(20),ETAT01(20),ETAT02(20),ETAT03(20),ETAT04(20)
      DIMENSION ETAT05(20),ETAT06(20),ETAT07(20),ETAT08(20),ETAT09(20)
      DIMENSION ETAT10(20),ETAT11(20)
      COMMON/NOGOF/NOGO,K,MATA,MATB
C ***
C *** ETAM DESIGN MOD - REMOVE 'NAMELIST' STATEMENTS AND SUBSTITUTE
C ***                   'FORMAT' STATEMENTS TO READ FROM EXC FILE,
C ***                   ONCE UNDER BASELINE CONDITIONS, ANOTHER WITH
C ***                   THE BENEFIT PATTERN APPLIED TO INPUT VARIABLES.
C ***
      NAMELIST/TECHF/ COURSE,N,ARATE,TLENGH,RCRATE,ARCYTM,WSCHOP,TSPOSD,
     1INTSPO,AMTSPO,EQTSPO,PURCHF,DEPF,LOFEQ,SQFTST,SQFTIN,SQFTAM,IMODEV
     2,PUIMD,UPDATE,SUPSPO,SUPSTD,MSCSPO,MSCSTD,PESP,TLEGTH,LOFFA,WSHOP1
      NAMELIST/COSTF/STUDSL,STCST1,STCST2,INSTSL,ADMSL,EQUNTC,PCTPCH,CPS
     1QFT,CIMD,EVIM,CSUPP,CMIS,DRATE,INRATE,FACST
      NAMELIST/DIMDF/GRAD,COPMT
C ***
C
C
C     PROGRAM INITIALIZATION
C
C
C ***
C *** ETAM DESIGN MOD - ETAM ENTRY POINT FOR START OF LOOP ONE.
C ***                   INITIALIZE LOOP INDICATOR (IND1).
    1 IND1  = 0
C ***
   10 AASIN = 0.0
      AAOB  = 0.0
      TINSTR= 0.0
      TADMIN= 0.0
      TEQUIP= 0.0
      TFACIL= 0.0
      TMDMNT= 0.0
      TSUPPL= 0.0
      TMISC = 0.0
      TPEQUP= 0.0
      TAEQUP= 0.0
      DATA COURSE /'NONE'/
```

```
TPEQUP= 0.0
AAEQUP= 0.0
PURCHF= 99999
DEPF   = 99999
RVEQ   = 0.0
RVFA   = 0.0
NSTUD  = 0.0
DSTUD  = 0.0
ISTUD  = 0.0
FSTUD  = 0.0
NTRAV  = 0.0
DTRAV  = 0.0
ITRAV  = 0.0
FTRAV  = 0.0
NINSTR= 0.0
DINSTR= 0.0
IINSTR= 0.0
FINSTR= 0.0
NADMIN= 0.0
DADMIN= 0.0
IADMIN= 0.0
FADMIN= 0.0
NFACIL= 0.0
DFACIL= 0.0
IFACIL= 0.0
FFACIL= 0.0
NSUPPL= 0.0
DSUPPL= 0.0
ISUPPL= 0.0
FSUPPL= 0.0
NMISC  = 0.0
DMISC  = 0.0
IMISC  = 0.0
FMISC  = 0.0
NEQUIP= 0.0
DEQUIP= 0.0
IEQUIP= 0.0
FEQUIP= 0.0
NIMD   = 0.0
DIMD   = 0.0
IIMD   = 0.0
FIMD   = 0.0
DAOB   = 0.0
SAOB   = 0.0
NOGO   = 0
IGRAD  = 0
IPSP   = 0
JAOB   = 0
ARATE  = 99999.
RCRATE= 99999.
ARCYTM= 99999.
WSCHOP= 99999.
WSHOP1= 99999.
TSPUSD= 99999.
PESP   = 99999.
INTSPO= 99999.
AMTSPO= 99999.
EQTSPO= 99999.
LOFEQ  = 99999
SQFTST= 99999.
SQFTIN= 99999.
SQFTAM= 99999.
IMODEV= 99999.
PUIMD  = 99999.
UPDATE= 99999.
SUPSPO= 99999.
SUPSTD= 99999.
MSCSPO= 99999.
MSCSTD= 99999.
STUDSL= 99999.
STCST1= 99999.
STCST2= 99999.
INVSTL= 99999.
ADMSL  = 99999.
EQUNTC= 99999.
PCTPCH= 99999.
CPSQFT= 99999.
CIMD   = 99999.
```

```
      EVIM   = 99999.
      CSUPP  = 99999.
      CMIS   = 99999.
      DRATE  = 99999.
      INRATE= 99999.
      N      = 99999.
      TLENGH= 99999.
      TLEGTH= 99999.
      FACST  = 99999.
      LOFFA  = 99999
      K      = 1
      DO 20 I=1,20
      CSTUD(I)  = 0.0
      CTRAV(I)  = 0.0
      CINSTR(I)= 0.0
      CADMIN(I)= 0.0
      CEQUIP(I)= 0.0
      CFACIL(I)= 0.0
      CIMDMT(I)= 0.0
      CSUPPL(I)= 0.0
      CMISC(I)  = 0.0
      COPTOT(I)= 0.0
      CGRAD(I)  = 0.0
      CSTIN(I)  = 0.0
      CSP(I)    = 0.0
      POEP(I)   = 0.0
      AEQUIP(I)= 0.0
      STUD(I)   = 0.0
      AOB(I)    = 0.0
      PSP(I)    = 0.0
      RINSTR(I)= 0.0
      RADMIN(I)= 0.0
      REQUIP(I)= 0.0
      PEQUIP(I)= 0.0
      AEQUIP(I)= 0.0
      RFACIL(I)= 0.0
      IMDMNT(I)= 0.0
      RSUPPL(I)= 0.0
      RMISC(I)  = 0.0
      AOBX(I)   = 0.0
      ETAGRD(I)  = 0.0
      ETASTU(I)  = 0.0
      ETAAUB(I)  = 0.0
      ETAINS(I)  = 0.0
      ETAADM(I)  = 0.0
      ETAEQP(I)  = 0.0
      ETAFAC(I)  = 0.0
      ETASUP(I)  = 0.0
      ETAMIS(I)  = 0.0
      ETAINV(I)  = 0.0
      ETAREC(I)  = 0.0
      GRAD(I)    = 99999.
      COPMT(I)   = 99999.
   20 CONTINUE
C
C     READ INPUT DATA
C
C
C  ***
C  *** ETAM DESIGN MOD - CHANGE 'READ' STATEMENTS TO READ FROM ORIGINAL
C  ***                   AND MODIFIED (WITH BENEFIT PATTERN) EXC FILE.
C  ***
      READ(5,TECHF)
      READ(5,COSTF)
      READ(5,DIMDF)
C  ***
C  *** ETAM DESIGN MOD - A LOGICAL '1' IN ITS RESPECTIVE POSITION
C  ***                   SELECTS A REPORT. THESE SHOULD BE SELECTABLE
C  ***                   INTERACTIVELY. THE  REPORT NAMES ARE:
C  ***                     RPT1 - INPUT TECHNICAL FACTORS
C  ***                     RPT2 - INPUT COST FACTORS
C  ***                     RPT3 - OUTPUT TECHNICAL RESULTS (TOTAL)
C  ***                     RPT4 - OUTPUT TECHNICAL RESULTS (ANNUAL)
C  ***                     RPT5 - OUTPUT COST RESULTS (TOTAL)
C  ***                     RPT6 - OUTPUT COST RESULTS (ANNUAL)
C  ***                     RPT7 - MISSING DATA ANALYSIS
C  ***                     LAST - INDICATES LAST INPUT STREAM DATA CASE
C  ***
      READ(5,6000) RPT1,RPT2,RPT3,RPT4,RPT5,RPT6,RPT7,LAST
```

```
 6000 FORMAT (8L1)
C ***
C
C     READ MISSING DATA ANALYSIS AND DEFAULT ROUTINES
C
C ***
C *** ETAM DESIGN MOD - EACH INPUT VARIABLE IS CHECKED FOR PRESENCE
C ***                   AND REASONABILITY. MISSING OR OUT-OF-RANGE
C ***                   DATA ARE RECORDED BY STATEMENT NUMBER BY THE
C ***                   'MISSNG SUBROUTINE'. VARIABLES WITH REASONABLE
C ***                   DEFAULT SUBSTITUTES HAVE A CRITICALITY CODE=1,
C ***                   VARIABLES WITH NO DEFAULT DATA HAVE CRIT CODE=0.
C ***                   FOLLOWING IS A LIST OF VARIABLES BY STATEMENT
C ***                   NUMER INCLUDING CRIT CODE AND PRESENT DEFAULT *
C ***                   VALUE.
C ***                   NO. VAR NAME       DESCRIPTION      DEFAULT   CRIT
C ***                    30  N        PLANNING PERIOD         20        1
C ***                    40  ARATE    ATTRITION RATE           0.0      1
C ***                    50  TLENGH   TRAINING LENGTH (WKS)    -        0
C ***                    55  TLEGTH   HRS/WK IN MEDIUM        40.0      1
C ***                    60  RCRATE   RECYCLE RATE          .S*ARATE    1
C ***                    90  WSCHOP   WKS/YR SCHOOL OPER      50.0      1
C ***                    95  WSHOPI   TRAINING WKS/YR         50.0      1
C ***                   100  TSPOSD   TIME STUD POS DOWN       0.0      1
C ***                   110  PESP     EXCESS STUD POSITIONS    0.10     1
C ***                   120  INTSPO   INST/STUD POS RATIO      0.0      1
C ***                   130  AMTSPO   ADMIN/STUD "     "       0.0      1
C ***                   140  EQTSPO   EQUIP/STUD "         CONDITIONAL 0/1
C ***                   160  PURCHF   EQUIP PURCHASE POLICY    1        1
C ***                   170  DEPF     EQUIP DEPRECIATION "     1        1
C ***                   175  LOFFA    LIFE OF FACLITY (YRS)    N        1
C ***                   180  LOFEQ    LIFE OF EQUIPMENT    CONDITIONAL 0/1
C ***                   200  SQFTST   SQ FT/STUDENT            0.0      1
C ***                   210  SQFTIN   SQ FT/INSTRUCTOR         0.0      1
C ***                   220  SQFTAM   SQ FT/ADMINISTRATIVE     0.0      1
C ***                   230  IMDDEV   INSTR MAT DEV RATIO  CONDITIONAL 0/1
C ***                   270  UPDATE   INSTR MAT MAINT      CONDITIONAL 0/1
C ***                   290  SUPSPO   SUPPLIES/STUD POS        0.0      1
C ***                   300  SUPSTD   SUPPLIES/STUDENT         0.0      1
C ***                   310  MSCSPO   MISC/STUDENT POS         0.0      1
C ***                   320  MSCSTD   MISC/STUDENT             0.0      1
C ***                   330  STUDSL   STUDENT SALARY           0.0      1
C ***                   340  STCST1   TRAVEL TO/FROM SCHOOL    0.0      1
C ***                   350  STCST2   TRAVEL DURING SCHOOL     0.0      1
C ***                   360  INSTSL   INSTRUCTOR SALARY        0.0      1
C ***                   370  ADMSL    ADMINISTRATIVE SALARY    0.0      1
C ***                   380  EQUNTC   EQUIP UNIT COST      CONDITIONAL 0/1
C ***                   400  PCTPCH   PCT MAX EQUIP PURCH      0.0      1
C ***                   410  CPSQFT   COST/SQ FT               0.0      1
C ***                   420  CIMD     COST/HR INSTR MAT DEV    0.0      1
C ***                   430  EVIM     ANNUAL MAINTENANCE PCT   0.0      1
C ***                   440  CSUPP    COST OF SUPPLIES         0.0      1
C ***                   450  CMIS     COST OF MISCELLANEOUS    0.0      1
C ***                   460  DRATE    DISCOUNT RATE            0.10     1
C ***                   470  INRATE   INFLATION RATE           0.08     1
C ***                   475  FACST    FACIL ACQ/REFURB COST    0.0      1
C ***                   480  GRAD     GRADUATES EACH YEAR      -        0
C ***                   490  COPMT    O&M COSTS EACH YEAR      0.0      1
C ***
   30 IF (N.GT.0.AND.N.LE.20) GO TO 40
      N     = 20
      CALL MISSNG (30,1)
   40 IF (ARATE.GE.0..AND.ARATE.LT.1.) GO TO 50
      ARATE = 0.0
      CALL MISSNG (40,1)
   50 IF (TLENGH.GT.0..AND.TLENGH.LT.200.) GO TO 55
      CALL MISSNG (50,0)
   55 IF (TLEGTH.GT.0..AND.TLEGTH.LT.121.) GO TO 60
      TLEGTH= 40.0
      CALL MISSNG (55,1)
   60 IF (RCRATE.GE.0..AND.RCRATE.LT.1.) GO TO 70
      RCRATE= .5*ARATE
      CALL MISSNG (60,1)
   70 IF (ARCYTM.GE.0..AND.ARCYTM.LT.200.) GO TO 90
      IF (RCRATE.EQ.0.) GO TO 80
      IF (TLENGH.LT.2.) GO TO 80
      CALL MISSNG (70,0)
      GO TO 90
 80 ARCYTM= 0.0
    CALL MISSNG (70,1)
 90 IF (WSCHOP.GT.0..AND.WSCHOP.LE.52.) GO TO 95
    WSCHOP= 50.0
    CALL MISSNG (90,1)
 95 IF (WSHOP1.GT.0..AND.WSHOP1.LE.150.) GO TO 100
    WSHOP1 = 50.0
    CALL MISSNG (95,1)
100 IF (TSPOSD.GE.0..AND.TSPOSD.LT.1.) GO TO 110
    TSPOSD= 0.0
    CALL MISSNG (100,1)
110 IF (PESP.GE.0..AND.PESP.LE.1.) GO TO 120
    PESP  = 0.1
    CALL MISSNG (110,1)
120 IF (INTSPO.GE.0..AND.INTSPO.LE.1.) GO TO 130
    INTSPO= 0.0
    CALL MISSNG (120,1)
130 IF (AMTSPO.GE.0..AND.AMTSPO.LE.1.) GO TO 140
    AMTSPO= 0.0
    CALL MISSNG (130,1)
140 IF (EQTSPO.GE.0..AND.EQTSPO.LE.1.) GO TO 160
    IF (EQUNTC.NE.99999..AND.EQUNTC.NE.0.) GO TO 150
    IF (LOFEQ.NE.99999.AND.LOFEQ.NE.0) GO TO 150
    EQTSPO= 0.0
    CALL MISSNG (140,1)
    GO TO 160
150 CALL MISSNG (140,0)
160 IF (PURCHF.GE.1.AND.PURCHF.LT.10) GO TO 170
    PURCHF= 1
    CALL MISSNG (160,1)
170 IF (DEPF.GE.1.AND.DEPF.LT.10) GO TO 175
    DEPF  = 1
    CALL MISSNG (170,1)
175 IF (LOFFA.GE.1.AND.LOFFA.LE.50) GO TO 180
    LOFFA = N
    CALL MISSNG (175,1)
180 IF (LOFEQ.GE.1.AND.LOFEQ.LE.50) GO TO 200
    IF (EQTSPO.NE.0.) GO TO 190
    IF (EQUNTC.NE.99999..AND.EQUNTC.NE.0.) GO TO 190
    LOFEQ = N
    CALL MISSNG (180,1)
    GO TO 200
190 CALL MISSNG (180,0)
200 IF (SQFTST.GE.0..AND.SQFTST.LT.1000.) GO TO 210
    SQFTST= 0.0
    CALL MISSNG (200,1)
210 IF (SQFTIN.GE.0..AND.SQFTIN.LT.1000.) GO TO 220
    SQFTIN= 0.0
    CALL MISSNG (210,1)
220 IF (SQFTAM.GE.0..AND.SQFTAM.LT.1000.) GO TO 230
    SQFTAM= 0.0
    CALL MISSNG (220,1)
230 IF (IMDDEV.GE.0..AND.IMDDEV.LT.1000.) GO TO 250
    IF (PUIMD.GT.0..AND.PUIMD.LE.1.) GO TO 240
    IF (UPDATE.GT.0..AND.UPDATE.LE.1.) GO TO 240
    IMDDEV= 0.0
    CALL MISSNG (230,1)
    GO TO 250
240 CALL MISSNG (230,0)
250 IF (PUIMD.GT.0..AND.PUIMD.LE.1.) GO TO 270
    IF (IMDDEV.NE.0.) GO TO 260
    IF (UPDATE.GT.0..AND.UPDATE.LE.1.) GO TO 260
    PUIMD = 1.0
    CALL MISSNG (250,1)
    GO TO 270
260 CALL MISSNG (250,0)
270 IF (UPDATE.GE.0..AND.UPDATE.LE.1.) GO TO 290
    IF (IMDDEV.NE.0.) GO TO 280
    IF (PUIMD.GT.0..AND.PUIMD.LT.1.) GO TO 280
    UPDATE= 0.0
    CALL MISSNG (270,1)
    GO TO 290
280 CALL MISSNG (270,0)
290 IF (SUPSPO.GE.0..AND.SUPSPO.LT.1000.) GO TO 300
    SUPSPO= 0.0
    CALL MISSNG (290,1)
300 IF (SUPSTD.GE.0..AND.SUPSTD.LT.1000.) GO TO 310
    SUPSTD= 0.0
```

```
      CALL MISSNG (300,1)
  310 IF (MSCSPO.GE.0..AND.MSCSPO.LT.1000.) GO TO 320
      MSCSPO= 0.0
      CALL MISSNG (310,1)
  320 IF (MSCSTD.GE.0..AND.MSCSTD.LT.1000.) GO TO 330
      MSCSTD= 0.0
      CALL MISSNG (320,1)
  330 IF (STUDSL.GE.0..AND.STUDSL.LT.50000.) GO TO 340
      STUDSL= 0.0
      CALL MISSNG (330,1)
  340 IF (STCST1.GE.0..AND.STCST1.LT.10000.) GO TO 350
      STCST1= 0.0
      CALL MISSNG (340,1)
  350 IF (STCST2.GE.0..AND.STCST2.LT.10000.) GO TO 360
      STCST2= 0.0
      CALL MISSNG (350,1)
  360 IF (INSTSL.GE.0..AND.INSTSL.LT.50000.) GO TO 370
      INSTSL= 0.0
      CALL MISSNG (360,1)
  370 IF (ADMSL.GE.0..AND.ADMSL.LT.50000.) GO TO 380
      ADMSL = 0.0
      CALL MISSNG (370,1)
  380 IF (EQUNTC.GE.0..AND.EQUNTC.NE.99999.) GO TO 400
      IF (EQTSPO.NE.0.) GO TO 390
      EQUNTC= 0.0
      CALL MISSNG (380,1)
      GO TO 400
  390 CALL MISSNG (380,0)
  400 IF (PCTPCH.GE.0..AND.PCTPCH.LE.1.) GO TO 410
      PCTPCH= 0.0
      CALL MISSNG (400,1)
  410 IF (CPSQFT.GE.0..AND.CPSQFT.NE.99999.) GO TO 420
      CPSQFT= 0.0
      CALL MISSNG (410,1)
  420 IF (CIMD.GE.0..AND.CIMD.LT.50000.) GO TO 430
      CIMD  = 0.0
      CALL MISSNG (420,1)
  430 IF (EVIM.GE.0..AND.EVIM.LT.1.) GO TO 440
      EVIM  = 0.0
      CALL MISSNG (430,1)
  440 IF (CSUPP.GE.0..AND.CSUPP.NE.99999.) GO TO 450
      CSUPP = 0.0
      CALL MISSNG (440,1)
  450 IF (CMIS.GE.0..AND.CMIS.NE.99999.) GO TO 460
      CMIS  = 0.0
      CALL MISSNG (450,1)
  460 IF (DRATE.GE.0..AND.DRATE.LT.1.) GO TO 470
      DRATE = 0.10
      CALL MISSNG (460,1)
  470 IF (INRATE.GE.0..AND.INRATE.LT.1.) GO TO 475
      INRATE= 0.08
      CALL MISSNG (470,1)
  475 IF (FACST.GE.0..AND.FACST.NE.99999.) GO TO 480
      FACST = 0.0
      CALL MISSNG (475,1)
  480 IF (GRAD(N).GE.0..AND.GRAD(N).NE.99999.) GO TO 490
      CALL MISSNG (480,0)
  490 IF (COPMT(N).GE.0..AND.COPMT(N).NE.99999.) GO TO 510
      DO 500 I=1,N
  500 COPMT(I) = 0.0
      CALL MISSNG (490,1)
C ***
  510 IF (NOGO.EQ.1) GO TO 960
      IF (N.GE.20) GO TO 518
      INDEX5    = N+1
      DO 515 I=INDEX5,20
      COPMT(I) = 0.0
  515 GRAD(I)  = 0
  518 CONTINUE
C
C     MODEL EQUATIONS - TECHNICAL FACTORS
C
      UIMDTM    = TLENGH*IMDDEV*PUIMD*TLEGTH
      CUIMD     = UIMDTM*CIMD
      RVIM      = CUIMD*EVIM
      IF (LOFFA.LE.N) GO TO 519
      RFVA = (LOFFA-N)*(FACST/LOFFA)
  519 DO 520 I=1,N
```

```
      STUD(I)   = GRAD(I)/(1.0-ARATE)
      IGRAD     = IGRAD+GRAD(I)
      AASIN     = AASIN+STUD(I)
      STUDMW(I)= (TLENGH*STUD(I))*(1.0-.5*ARATE)
      SMWRRC(I)= RCRATE*STUD(I)*ARCYTM
      AOB(I)    = (STUDMW(I)+SMWRRC(I))/WSCHOP
      DAOB      = DAOB + (AOB(I)-AOBX(I))
      SAOB      = SAOB + AOB(I)
      AAOB      = AAOB+AOB(I)
      PSP(I)     = (((AOB(I)*WSHOP1/WSCHOP)/(1.0-TSPOSD))*(1.0+PESP))
      IPSP      = IPSP+PSP(I)
      RINSTR(I)= PSP(I)*INTSPO
      TINSTR    = TINSTR+RINSTR(I)
      RADMIN(I)= PSP(I)*AMTSPO
      TADMIN    = TADMIN+RADMIN(I)
      REQUIP(I)= PSP(I)*EQTSPO
      TEQUIP    = TEQUIP+REQUIP(I)
      RFACIL(I)= PSP(I)*(SQFTST+INTSPO*SQFTIN+AMTSPO*SQFTAM)
      TFACIL    = TFACIL+RFACIL(I)
      IMDMNT(I)= UIMDTM*UPDATE
      TMDMNT    = TMDMNT+IMDMNT(I)
      RSUPPL(I)= PSP(I)*SUPSPO+STUD(I)*SUPSTD
      TSUPPL    = TSUPPL+RSUPPL(I)
      RMISC(I) = PSP(I)*MSCSPO+STUD(I)*MSCSTD
      JAOB      = JAOB+AOBX(I)
      TMISC     = TMISC+RMISC(I)
C
C     COURSE TECHNICAL FACTORS TO ACCUMULATE FOR ETAM CMR FILE
C
      ETAGRD(I) = GRAD(I)
      ETASTU(I) = STUD(I)
      ETAAOB(I) = AOB(I)
      ETAINS(I) = RINSTR(I)
      ETAADM(I) = RADMIN(I)
      ETAEQP(I) = REQUIP(I)
      ETAFAC(I) = RFACIL(I)
      ETASUP(I) = RSUPPL(I)
  520 ETAMIS(I) = RMISC(I)
      IF (DAOB.LT.0.) DAOB=DAOB*(-1.)
      IAOB      = ((DAOB/SAOB)+.5)*100
      IF (IAOB.GT.5.AND.JAOB.NE.0) CALL MISSNG (520,IAOB)
      AASIN     = AASIN/N
      AGRAD     = IGRAD/N
      AAOB      = AAOB/N
      APSP      = IPSP/N
      AINSTR    = TINSTR/N
      AADMIN    = TADMIN/N
      AEQUP     = TEQUIP/N
      AFACIL    = TFACIL/N
      AIMDMT    = TMDMNT/N
      ASUPPL    = TSUPPL/N
      AMISC     = TMISC/N
C
C     THIS ROUTINE CALCULATES EQUIPMENT PURCHASE AND DEPRECIATION
C     SCHEDULES FROM USER SELECTED POLICY OPTIONS
C
C                       - PURCHASE POLICIES ANALYSIS
C
      GO TO (530,540,550,560,570,580,590,600,610),PURCHF
C
C                       POLICY 1 - PURCHASE TOTAL AVERAGE RQMTS
C                                  IN FIRST YEAR OF EQUIP LIFE
  530 CONTINUE
      DO 534 I=1,N,LOFEQ
      TEMP1 = 0.0
      INDEX1= LOFEQ+I-1
      DO 532 J=I,INDEX1
      TEMP1 = REQUIP(J)+TEMP1
  532 CONTINUE
      PEQUIP(I)= TEMP1/LOFEQ
  534 CONTINUE
      GO TO 620
C                       POLICY 2 - PURCHASE TOTAL PERCENT OF MAXIMUM
C                                  RQMTS IN FIRST YEAR OF EQUIP LIFE
  540 CONTINUE
      DO 544 I=1,N,LOFEQ
      TEMP1 = 0.0
      INDEX1= LOFEQ+I-1
      DO 542 J=I,INDEX1
      IF (TEMP1.LT.REQUIP(J)) TEMP1=REQUIP(J)
  542 CONTINUE
      PEQUIP(I)= TEMP1*PCTPCH
  544 CONTINUE
      GO TO 620
C                         POLICY 3 - PURCHASE UP TO AVERAGE RQMTS BUT
C                                    NOT EXCEEDING RQMTS IN YR OF PURCH
  550 CONTINUE
      DO 558 I=1,N,LOFEQ
      TEMP1 = 0.0
      INDEX1= LOFEQ+I-1
      DO 552 J=I,INDEX1
      TEMP1 = REQUIP(J)+TEMP1
  552 CONTINUE
      PEQUIP(I)= TEMP1/LOFEQ
      IF (PEQUIP(I).LE.REQUIP(I)) GO TO 558
      TEMP1 = PEQUIP(I)
      INDEX1= LOFEQ+I-1
      DO 556 J=I,INDEX1
      IF (TEMP1.LE.0.) GO TO 556
      IF (TEMP1.GT.REQUIP(J)) PEQUIP(J)=REQUIP(J)
      IF (TEMP1.LE.REQUIP(J)) PEQUIP(J)=TEMP1
      TEMP1 = TEMP1-PEQUIP(J)
  556 CONTINUE
  558 CONTINUE
      GO TO 620
C                         POLICY 4 - PURCHASE UP TO PCT OF MAX RQMTS BUT
C                                    NOT EXCEEDING RQMTS IN YR OF PURCH
  560 CONTINUE
      DO 568 I=1,N,LOFEQ
      TEMP1 = 0.0
      INDEX1= LOFEQ+I-1
      DO 562 J=I,INDEX1
      IF (TEMP1.LT.REQUIP(J)) TEMP1=REQUIP(J)
  562 CONTINUE
      PEQUIP(I)= TEMP1*PCTPCH
      IF (PEQUIP(I).LE.REQUIP(I)) GO TO 568
      TEMP1 = PEQUIP(I)
      INDEX1= LOFEQ+I-1
      DO 566 J=I,INDEX1
      IF (TEMP1.LE.0.) GO TO 566
      IF (TEMP1.GT.REQUIP(J)) PEQUIP(J)=REQUIP(J)
      IF (TEMP1.LE.REQUIP(J)) PEQUIP(J)=TEMP1
      TEMP1 = TEMP1-PEQUIP(J)
  566 CONTINUE
  568 CONTINUE
      GO TO 620
  570 GO TO 530
  580 GO TO 530
  590 GO TO 530
  600 GO TO 530
  610 GO TO 530
C
C                       - DEPRECIATION POLICIES ANALYSIS
C
  620 GO TO (630,640,650,660,670,680,690,700,710) , DEPF
C
C                         POLICY 1 - DEPRECIATE LINEARLY OVER
C                                    LIFE OF EQUIPMENT
  630 CONTINUE
      TEMP2     = LOFEQ
      DO 632 I=1,LOFEQ
      TEMP1     = I
      PDEP(I)   = TEMP1/TEMP2
  632 CONTINUE
      GO TO 730
C                         POLICY 2 - DEPRECIATE BY SUM-OF-DIGITS
C                                    OVER LIFE OF EQUIPMENT
  640 CONTINUE
      SUM       = 0.0
      DO 642 I=1,LOFEQ
      SUM       = SUM+I
  642 CONTINUE
      PDEP(1)   = (LOFEQ)/SUM
      DO 644 I=2,LOFEQ
      PDEP(I)   = (LOFEQ-I)/SUM+PDEP(I-1)
  644 CONTINUE
```

```
      GO TO 730
C                                  POLICY 3 - SET DEPRECIATION SCHEDULE
C                                             TO FULLY DEPRECIATED ALL YEARS
  650 CONTINUE
      DO 652 I=1,LOFEQ
      PDEP(I)   = 0.0
  652 CONTINUE
      GO TO 730
  660 GO TO 630
  670 GO TO 630
  680 GO TO 630
  690 GO TO 630
  700 GO TO 630
  710 GO TO 630
C
C                              - COST FACTORS
C
  730 L         = N+1
      IF (L.GT.20) GO TO 750
      DO 740 I=L,20
      GRAD(I)   = 0
  740 COPMT(L) = 0.0
  750 CONTINUE
      TOTAEQ    = 0.0
      DO 790 I=1,N
      CSTUD(I) =  ((STUDMW(I)+SMWRRC(I))/52)*STUDSL
      CTRAV(I) =  STUD(I)*(STCST1+STCST2*(1.0-.5*ARATE))
      CINSTR(I)=  RINSTR(I)*INSTSL
      CADMIN(I)=  RADMIN(I)*ADMSL
      TPEQUP    = TPEQUP+PEQUIP(I)
      INDEX4= PEQUIP(I)+0.5
      PEQUIP(I)= INDEX4
      CEQUIP(I)= PEQUIP(I)*EQUNTC
      LIFERV    = I+LOFEQ-N-1
      IF (LIFERV.LE.0) GO TO 760
      RVEQ      = RVEQ+CEQUIP(I)*PDEP(LIFERV)
  760 CONTINUE
      TOTAEQ    = TOTAEQ+PEQUIP(I)
      AEQUIP(I)= TOTAEQ
      TAEQUP    = TAEQUP+AEQUIP(I)
      LOBSLT    = I-LOFEQ
      IF (LOBSLT.LE.0) GO TO 770
      TOTAEQ    = TOTAEQ-PEQUIP(LOBSLT)
      AEQUIP(I)= TOTAEQ
  770 CSUPPL(I) = RSUPPL(I)*CSUPP
      CMISC(I)  = RMISC(I)*CMIS
      CINV1 = CEQUIP(I)
      CREC1 = AEQUIP(I)*COPMT(I)
      CREC2 = RFACIL(I)*CPSQFT
      CREC3 = IMDMNT(I)*CIMD
      CRECT = CREC1 + CREC2 + CREC3 + CSTUD(I) + CTRAV(I) + CINSTR(I) +
     1        CADMIN(I) + CSUPPL(I) + CMISC(I)
C
C     COURSE COST FACTORS TO ACCUMULATE FOR ETAM CMR FILE
C
C                         - FIRST YEAR
C
      IF (I.NE.1) GO TO 771
      ETAINV(1) = CINV1 + FACST + CUIMD
      CFACIL(I) = CREC2 + FACST
      CIMDMT(I) = CREC3 + CUIMD
  771 IF (I.EQ.1) GO TO 772
C
C                         - REMAINING YEARS
C
      ETAINV(I) = CINV1
      CFACIL(I) = CREC2
      CIMDMT(I) = CREC3
C
  772 ETAREC(I) = CRECT
      CEQUIP(I) = CINV1 + CREC1
      IF (I.NE.N) GO TO 773
      ETAINV(N) = ETAINV(N) - RVEQ - RFVA - RVIM
  773 CONTINUE
C
C                         - COST SUMMARIZATION
C
      DISCNT(I)= (2.0+DRATE)/(2.0*(1.0+DRATE)**I)
      INFLAT(I)= (2.0*(1.0+INRATE)**I)/(2.0+INRATE)
      NSTUD     = NSTUD+CSTUD(I)
      DSTUD     = DSTUD+CSTUD(I)*DISCNT(I)
      ISTUD     = ISTUD+CSTUD(I)*INFLAT(I)
      FSTUD     = FSTUD+CSTUD(I)*INFLAT(I)*DISCNT(I)
      NTRAV     = NTRAV+CTRAV(I)
      DTRAV     = DTRAV+CTRAV(I)*DISCNT(I)
      ITRAV     = ITRAV+CTRAV(I)*INFLAT(I)
      FTRAV     = FTRAV+CTRAV(I)*INFLAT(I)*DISCNT(I)
      NINSTR    = NINSTR+CINSTR(I)
      DINSTR    = DINSTR+CINSTR(I)*DISCNT(I)
      IINSTR    = IINSTR+CINSTR(I)*INFLAT(I)
      FINSTR    = FINSTR+CINSTR(I)*INFLAT(I)*DISCNT(I)
      NADMIN    = NADMIN+CADMIN(I)
      DADMIN    = DADMIN+CADMIN(I)*DISCNT(I)
      IADMIN    = IADMIN+CADMIN(I)*INFLAT(I)
      FADMIN    = FADMIN+CADMIN(I)*INFLAT(I)*DISCNT(I)
      NSUPPL    = NSUPPL+CSUPPL(I)
      DSUPPL    = DSUPPL+CSUPPL(I)*DISCNT(I)
      ISUPPL    = ISUPPL+CSUPPL(I)*INFLAT(I)
      FSUPPL    = FSUPPL+CSUPPL(I)*INFLAT(I)*DISCNT(I)
      NMISC     = NMISC+CMISC(I)
      DMISC     = DMISC+CMISC(I)*DISCNT(I)
      IMISC     = IMISC+CMISC(I)*INFLAT(I)
      FMISC     = FMISC+CMISC(I)*INFLAT(I)*DISCNT(I)
      IF  (I.NE.N) GO TO 780
      CEQUIP(I)=CEQUIP(I)-RVEQ
      CIMDMT(I)=CIMDMT(I)-RVIM
  780 NEQUIP    = NEQUIP+CEQUIP(I)
      CFACIL(I) = CFACIL(I)-RFVA
      DEQUIP    = DEQUIP+CEQUIP(I)*DISCNT(I)
      IEQUIP    = IEQUIP+CEQUIP(I)*INFLAT(I)
      FEQUIP    = FEQUIP+CEQUIP(I)*INFLAT(I)*DISCNT(I)
      NIMD      = NIMD+CIMDMT(I)
      DIMD      = DIMD+CIMDMT(I)*DISCNT(I)
      IIMD      = IIMD+CIMDMT(I)*INFLAT(I)
      FIMD      = FIMD+CIMDMT(I)*DISCNT(I)*INFLAT(I)
      NFACIL    = NFACIL+CFACIL(I)
      DFACIL    = DFACIL+CFACIL(I)*DISCNT(I)
      IFACIL    = IFACIL+CFACIL(I)*INFLAT(I)
      FFACIL    = FFACIL+CFACIL(I)*INFLAT(I)*DISCNT(I)
      COPTOT(I)= CSTUD(I)+CTRAV(I)+CINSTR(I)+CADMIN(I)+CFACIL(I)
     1          +CIMDMT(I)+CSUPPL(I)+CMISC(I)+CEQUIP(I)
      CGRAD(I) = COPTOT(I)/GRAD(I)
      CSTIN(I) = COPTOT(I)/STUD(I)
  790 CSP(I)   = COPTOT(I)/PSP(I)
      APEQUP    = TPEQUP/N
      AAEQUP    = TAEQUP/N
      SNAOC=NSTUD+NTRAV+NINSTR+NADMIN+NFACIL+NSUPPL+NMISC+NEQUIP+NIMD
      SDAOC=DSTUD+DTRAV+DINSTR+DADMIN+DFACIL+DSUPPL+DMISC+DEQUIP+DIMD
      SIAOC=ISTUD+ITRAV+IINSTR+IADMIN+IFACIL+ISUPPL+IMISC+IEQUIP+IIMD
      SFAOC=FSTUD+FTRAV+FINSTR+FADMIN+FFACIL+FSUPPL+FMISC+FEQUIP+FIMD
      GNAOC=SNAOC/IGRAD
      GDAOC=SDAOC/IGRAD
      GIAOC=SIAOC/IGRAD
      GFAOC=SFAOC/IGRAD
      UNAOC=SNAOC/(AASIN*N)
      UDAOC=SDAOC/(AASIN*N)
      UIAOC=SIAOC/(AASIN*N)
      UFAOC=SFAOC/(AASIN*N)
      PNAOC=SNAOC/IPSR
      PDAOC=SDAOC/IPSP
      PIAOC=SIAOC/IPSP
      PFAOC=SFAOC/IPSP
      DRATE = DRATE*100.
      INRATE= INRATE*100.
C
C     WRITE INPUT TECHNICAL FACTORS
C
  900 IF (.NOT.RPT1) GO TO 910
      WRITE (6,1000) COURSE,DRATE,N,INRATE,(GRAD(I),I=1,10),
     1               (GRAD(I),I=11,20),ARATE,TSPOSD,TLENGH,PESP,RCRATE,
     2               EQTSPO
 1000 FORMAT ('1',28X,'COURSE COST ANALYSIS',/,1X,'INPUT:',/,3X,
     1        'COURSE NUMBER    - ',A4,25X,'DISCOUNT RATE   - ',F4.1,
     2        ' PERCENT',/,3X,'PLANNING PERIOD - ',I2,27X,
     3        'INFLATION RATE - ',F4.1,' PERCENT',//,1X,
     4        'INPUT TECHNICAL FACTORS:',/,3X,'PLNG YR','    1       2',
     5         '      3        4         5         6         7         8         9       ',
     6         '10',/,3X,'GRADS-',10(1X,I6),/,3X,'PLNG YR','  11        ',
     7         '12      13        14        15        16        17        18        19',
     8         '        20',/,3X,'GRADS-',10(1X,I6),//,3X,
     9         'ATTRITION RATE (PCT)     -      ',F4.2,8X,
     1         'TIME STUD POS DOWN (PCT) -      ',F3.2,/,3X,
     A         'TRAINING LENGTH (WKS)    - ',F5.1,9X,
     B         'PCT EXTRA STUD POSITIONS -      ',F3.2,/,3X,
     C         'RECYCLE RATE (PCT)       -      ',F3.2,8X,
     D         'EQUIP/STUD POS RATIO     -      ',F4.2)
      WRITE (6,1010) ARCYTM,TLEGTH,WSCHOP,IMDDEV,INTSPO,PUIMD,
     3               AMTSPO,UPDATE,SQFTST,EVIM,SQFTIN,SUPSPO,SQFTAM,
     4               SUPSTD,PURCHF,MSCSPO,DEPF,MSCSTD,WSHOP1
 1010 FORMAT (3X,
     E         'AVE. RECYCLE TIME (WKS)-  ',F4.1,9X,
     F         'TIME IN MEDIUM               - ',F5.1,/,3X,
     G         'WEEKS SCHOOL OPERATES   -  ',F4.1,9X,
     H         'COURSE DEV HRS/COURSE HR - ',F5.1,/,3X,
     I         'INSTR/STUD POS RATIO     -  ',F5.2,8X,
     J         'PCT COURSE REQUIRING DEV.-    ',F4.2,/,3X,
     K         'ADMIN/STUD POS RATIO     -   ',F4.3,7X,
     L         'PCT INSTR MAT. MAINTAINED-     ',F3.2,/,3X,
     M         'SQ FEET/STUDENT POS.    - ',F6.2,8X,
     N         'PCT INSTR MAT. RMNG VALUE-     ',F3.2,/,3X,
     O         'SQ FEET/INSTR. POS.     - ',F6.2,8X,
     P         'SUPPLIES/STUDENT POSITION- ',F5.1,/,3X,
     Q         'SQ FEET/ADMIN POS.      - ',F6.2,8X,
     R         'SUPPLIES/STUDENT             - ',F5.1,/,3X,
     S         'PURCHASE POLICY          -     ',I1,11X,
     T         'MISC/STUDENT POSITION     - ',F5.1,/,3X,
     U         'DEPRECIATION POLICY      -     ',I1,11X,
     V         'MISC/STUDENT                 - ',F5.1,/,3X,
     W         'WEEKS SCHOOL AVAILABLE -   ',F4.1)
C
C     WRITE INPUT COST FACTORS
C
  910 IF (.NOT.RPT2) GO TO 920
      WRITE (6,1001) COURSE,DRATE,N,INRATE,(COPMT(I),I=1,10),
     1               (COPMT(I),I=11,20),STUDSL,CIMD,STCST1,CSUPP,STCST2,
     2               CMIS,INSTSL,PCTPCH,ADMSL,CPSQFT
 1001 FORMAT ('1',28X,'COURSE COST ANALYSIS',/,1X,'INPUT:',/,3X,
     1         'COURSE NUMBER    - ',A4,25X,'DISCOUNT RATE   - ',F4.1,
     2         ' PERCENT',/,3X,'PLANNING PERIOD - ',I2,27X,
     3         'INFLATION RATE - ',F4.1,' PERCENT',//,1X,
     4         'INPUT COST FACTORS:',/,3X,'PLNG YR','     1         2',
     5         '      3        4         5         6         7         8         9       ',
     6         '10',/,3X,'O&MN- ',10(1X,F6.0),/,3X,'PLNG YR','  11',
     7         '        12       13        14        15        16        17        18       ',
     8         '19        20',/,3X,'O&MN- ',10(1X,F6.0),//,3X,
     9         'STUDENT SALARY (ANNUAL)- ',F8.2,6X,
     1         'COST/HR OF INSTR MAT DEV      - ',F6.2,/,3X,
     A         'STUD TRAVEL TO/FROM      -   ',F7.2,6X,
     B         'SUPPLIES COST                 - ',F6.2,/,3X,
     C         'STUD TRAVEL IN COURSE  -   ',F7.2,6X,
     D         'MISCELLANEOUS COST            - ',F6.2,/,3X,
     E         'INSTR SALARY (ANNUAL)  - ',F8.2,6X,
     F         'PCT MAX EQUIP PURCHASED       -    ',F4.2,/,3X,
     G         'ADMIN SALARY (ANNUAL)  - ',F8.2,6X,
     H         'FACILITY COST/SQ FOOT         - ',F6.2)
      WRITE (6,1007) EQUNTC,LOFEQ,FACST,LOFFA
 1007 FORMAT (3X,
     I         'EQUIPMENT UNIT COST     -',F9.2,6X,
     J         'LIFE OF EQUIPMENT             - ',I2,/,3X,
     K         'FACIL INIT/REFURB COST -',F9.2,6X,
     L         'LIFE OF FACILITY              - ',I2)
C
C     WRITE SUMMARY OF ANNUAL TECHNICAL RESULTS
C
  920 IF (.NOT.RPT3) GO TO 930
      WRITE (6,1003) COURSE,DRATE,N,INRATE,AGRAD,AASIN,AADB,APSP,
     1               AINSTR,AADMIN,AEQUP,APEQUP,AAEQUP,AFACIL,AIMDMT,
     2               ASUPPL,AMISC
 1003 FORMAT ('1',28X,'COURSE COST ANALYSIS',/,1X,'INPUT:',/,3X,
     A         'COURSE NUMBER    - ',A4,25X,'DISCOUNT RATE   - ',F4.1,
     B         ' PERCENT',/,3X,'PLANNING PERIOD - ',I2,27X,
     C         'INFLATION RATE - ',F4.1,' PERCENT',//,1X,
     D         'OUTPUT TECHNICAL RESULTS:',//,3X,
     E         'AVE. NO. OF GRADUATES RQD        - ',F10.1,/,3X,
     F          'AVE. NO. OF STUDENTS IN          - ',F10.1,/,3X,
     G          'AVE. AVERAGE ON BOARD            - ',F10.1,/,3X,
     H          'AVE. NO. OF STUDENT POSITIONS    - ',F10.1,/,3X,
     I          'AVE. NO. OF INSTRUCTORS RQD      - ',F10.1,/,3X,
     J          'AVE. NO. OF ADMIN PERS RQD       - ',F10.1,/,3X,
     K          'AVE. NO. OF EQUIPMENTS RQD       - ',F10.1,/,3X,
     L          'AVE. ANNUAL EQUIP PURCHASES      - ',F10.1,/,3X,
     M          'AVE. ANNUAL EQUIP AVAILABLE      - ',F10.1,/,3X,
     N          'AVE. NO. SQUARE FEET RQD         - ',F10.1,/,3X,
     O          'AVE. INSTR. MAT. MAINT HR RQD    - ',F10.1,/,3X,
     P          'AVE. NO. SUPPLIES RQD            - ',F10.1,/,3X,
     Q          'AVE. NO. MISCELLANEOUS RQD       - ',F10.1)
C
C     WRITE ANNUAL TECHNICAL RESULTS
C
  930 IF (.NOT.RPT4) GO TO 940
      WRITE (6,1005) COURSE,DRATE,N,INRATE,(DIGIT(I),I=1,5),
     1               (GRAD(I),I=1,5),(STUD(I),I=1,5),
     2               (AOB(I),I=1,5),(PSP(I),I=1,5),
     3               (RINSTR(I),I=1,5),(RADMIN(I),I=1,5),
     4               (REQUIP(I),I=1,5),(PEQUIP(I),I=1,5),
     5               (AEQUIP(I),I=1,5),(RFACIL(I),I=1,5),
     6               (IMDMNT(I),I=1,5),(RSUPPL(I),I=1,5),
     7               (RMISC(I),I=1,5)
      IF (N.LE.5) GO TO 940
      WRITE (6,1005) COURSE,DRATE,N,INRATE,(DIGIT(I),I=6,10),
     1               (GRAD(I),I=6,10),(STUD(I),I=6,10),
     2               (AOB(I),I=6,10),(PSP(I),I=6,10),
     3               (RINSTR(I),I=6,10),(RADMIN(I),I=6,10),
     4               (REQUIP(I),I=6,10),(PEQUIP(I),I=6,10),
     5               (AEQUIP(I),I=6,10),(RFACIL(I),I=6,10),
     6               (IMDMNT(I),I=6,10),(RSUPPL(I),I=6,10),
     7               (RMISC(I),I=6,10)
      IF (N.LE.10) GO TO 940
      WRITE (6,1005) COURSE,DRATE,N,INRATE,(DIGIT(I),I=11,15),
     1               (GRAD(I),I=11,15),(STUD(I),I=11,15),
     2               (AOB(I),I=11,15),(PSP(I),I=11,15),
     3               (RINSTR(I),I=11,15),(RADMIN(I),I=11,15),
     4               (REQUIP(I),I=11,15),(PEQUIP(I),I=11,15),
     5               (AEQUIP(I),I=11,15),(RFACIL(I),I=11,15),
     6               (IMDMNT(I),I=11,15),(RSUPPL(I),I=11,15),
     7               (RMISC(I),I=11,15)
      IF (N.LE.15) GO TO 940
      WRITE (6,1005) COURSE,DRATE,N,INRATE,(DIGIT(I),I=16,20),
     1               (GRAD(I),I=16,20),(STUD(I),I=16,20),
     2               (AOB(I),I=16,20),(PSP(I),I=16,20),
     3               (RINSTR(I),I=16,20),(RADMIN(I),I=16,20),
     4               (REQUIP(I),I=16,20),(PEQUIP(I),I=16,20),
     5               (AEQUIP(I),I=16,20),(RFACIL(I),I=16,20),
     6               (IMDMNT(I),I=16,20),(RSUPPL(I),I=16,20),
     7               (RMISC(I),I=16,20)
 1005 FORMAT ('1',28X,'COURSE COST ANALYSIS',/,1X,'INPUT:',/,3X,
     1          'COURSE NUMBER    - ',A4,25X,'DISCOUNT RATE    - ',F4.1,
     2          ' PERCENT',/,3X,'PLANNING PERIOD - ',I2,27X,
     3          'INFLATION RATE - ',F4.1,' PERCENT',//,1X,
     4          'OUTPUT TECHNICAL RESULTS:',//,32X,
     5          'YEAR ',I2,'    YEAR ',I2,'    YEAR ',I2,'    YEAR ',I2,
     6          '    YEAR ',I2,//,3X,
     7          'NO. OF GRADUATES RQD        ',5(4X,I6),/,3X,
     8          'NO. OF STUDENTS IN          ',5(1X,F9.0),/,3X,
     9          'AVERAGE ON BOARD            ',5(1X,F9.0),/,3X,
     A          'NO. OF STUDENT POSITIONS    ',5(1X,F9.0),/,3X,
     B          'NO. OF INSTRUCTORS RQD      ',5(1X,F9.0),/,3X,
     C          'NO. OF ADMIN PERS RQD       ',5(1X,F9.0),/,3X,
     D          'NO. OF EQUIPMENTS RQD       ',5(1X,F9.0),/,3X,
     E          'ANNUAL EQUIP PURCHASES      ',5(1X,F9.0),/,3X,
     F          'ANNUAL EQUIP AVAILABLE      ',5(4X,I6),/,3X,
     G          'NO. SQUARE FEET RQD         ',5(1X,F9.0),/,3X,
     H          'NO. INSTR MAT MAINT HR RQD',5(1X,F9.0),/,3X,
     I          'NO. SUPPLIES RQD            ',5(1X,F9.0),/,3X,
     J          'NO. MISCELLANEOUS RQD       ',5(1X,F9.0))
C
C     WRITE SUMMARY OF ANNUAL COST RESULTS
C
  940 IF (.NOT.RPT5) GO TO 950
      WRITE (6,1002) COURSE,DRATE,N,INRATE,NSTUD,DSTUD,ISTUD,FSTUD,
     1               NTRAV,DTRAV,ITRAV,FTRAV,NINSTR,DINSTR,IINSTR,
     2               FINSTR,NADMIN,DADMIN,IADMIN,FADMIN,NEQUIP,DEQUIP,
     3               IEQUIP,FEQUIP,NFACIL,DFACIL,IFACIL,FFACIL,NIMD,
     4               DIMD,IIMD,FIMD,NSUPPL,DSUPPL,ISUPPL,FSUPPL,NMISC,
     5               DMISC,IMISC,FMISC,SNAOC,SDAOC,SIAOC,SFAOC,GNAOC,
     6               GDAOC,GIAOC,GFACC,UNAOC,UDAOC,UIAOC,UFAOC,PNAOC,
     7               PDAOC,PIAOC,PFAOC
 1002 FORMAT ('1',28X,'COURSE COST ANALYSIS',/,1X,'INPUT:',/,3X,
     1        'COURSE NUMBER    - ',A4,25X,'DISCOUNT RATE   - ',F4.1,
     2        ' PERCENT',/,3X,'PLANNING PERIOD - ',I2,27X,
     3        'INFLATION RATE - ',F4.1,' PERCENT',//,1X,
     A        'OUTPUT COST RESULTS:',10X,
     4        'NON-DISC     DISCOUNTED  NON-DISC     DISCOUNTED',/,31X,
     5        'CONSTANT $  CONSTANT $  CURRENT $   CURRENT $',/,3X,
     6        'STUDENT SALARIES',10X,4(2X,F10.0),/,3X,
     7        'STUDENT TRAVEL',12X,4(2X,F10.0),/,3X,
     8        'INSTRUCTOR SALARY',9X,4(2X,F10.0),/,3X,
     9        'ADMINISTRATIVE SALARY',5X,4(2X,F10.0),/,3X,
     1        'EQUIPMENT',17X,4(1X,F11.0),/,3X,
     A        'FACILITIES',16X,4(1X,F11.0),/,3X,
     B        'INSTRUCTIONAL MATERIALS',3X,4(2X,F10.0),/,3X,
     C        'SUPPLIES',18X,4(2X,F10.0),/,3X,
     D        'MISCELLANEOUS',13X,4(2X,F10.0),//,3X,
     E        'TOTAL OPERATIONAL',9X,4(1X,F11.0),//,3X,
     F        'COST PER GRADUATE',8X,4(4X,F8.2),/,3X,
     G        'COST PER STUDENT INPUT',3X,4(4X,F8.2),/,3X,
     H        'COST PER STUDENT POSITION',4(4X,F8.2))
C
C     WRITE ANNUAL COST RESULTS
C
  950 IF (.NOT.RPT6) GO TO 960
      WRITE (6,1004) COURSE,DRATE,N,INRATE,(DIGIT(I),I=1,5),
     1               (CSTUD(I),I=1,5),(CTRAV(I),I=1,5),
     2               (CINSTR(I),I=1,5),(CADMIN(I),I=1,5),
     3               (CEQUIP(I),I=1,5),(CFACIL(I),I=1,5),
     4               (CIMDMT(I),I=1,5),(CSUPPL(I),I=1,5),
     5               (CMISC(I),I=1,5),(COPTOT(I),I=1,5),
     6               (CGRAD(I),I=1,5),(CSTIN(I),I=1,5),
     7               (CSP(I),I=1,5)
      IF (N.LE.5) GO TO 960
      WRITE (6,1004) COURSE,DRATE,N,INRATE,(DIGIT(I),I=6,10),
     1               (CSTUD(I),I=6,10),(CTRAV(I),I=6,10),
     2               (CINSTR(I),I=6,10),(CADMIN(I),I=6,10),
     3               (CEQUIP(I),I=6,10),(CFACIL(I),I=6,10),
     4               (CIMDMT(I),I=6,10),(CSUPPL(I),I=6,10),
     5               (CMISC(I),I=6,10),(COPTOT(I),I=6,10),
     6               (CGRAD(I),I=6,10),(CSTIN(I),I=6,10),
     7               (CSP(I),I=6,10)
      IF (N.LE.10) GO TO 960
      WRITE (6,1004) COURSE,DRATE,N,INRATE,(DIGIT(I),I=11,15),
     1               (CSTUD(I),I=11,15),(CTRAV(I),I=11,15),
     2               (CINSTR(I),I=11,15),(CADMIN(I),I=11,15),
     3               (CEQUIP(I),I=11,15),(CFACIL(I),I=11,15),
     4               (CIMDMT(I),I=11,15),(CSUPPL(I),I=11,15),
     5               (CMISC(I),I=11,15),(COPTOT(I),I=11,15),
     6               (CGRAD(I),I=11,15),(CSTIN(I),I=11,15),
     7               (CSP(I),I=11,15)
      IF (N.LE.15) GO TO 960
      WRITE (6,1004) COURSE,DRATE,N,INRATE,(DIGIT(I),I=16,20),
     1               (CSTUD(I),I=16,20),(CTRAV(I),I=16,20),
     2               (CINSTR(I),I=16,20),(CADMIN(I),I=16,20),
     3               (CEQUIP(I),I=16,20),(CFACIL(I),I=16,20),
     4               (CIMDMT(I),I=16,20),(CSUPPL(I),I=16,20),
     5               (CMISC(I),I=16,20),(COPTOT(I),I=16,20),
     6               (CGRAD(I),I=16,20),(CSTIN(I),I=16,20),
     7               (CSP(I),I=16,20)
 1004 FORMAT ('1',28X,'COURSE COST ANALYSIS',/,1X,'INPUT:',/,3X,
     1        'COURSE NUMBER    - ',A4,25X,'DISCOUNT RATE   - ',F4.1,
     2        ' PERCENT',/,3X,'PLANNING PERIOD - ',I2,27X,
     3        'INFLATION RATE - ',F4.1,' PERCENT',//,1X,
     4        'OUTPUT COST RESULTS:   (NON-DISC CONSTANT $)',//,32X,
     5        'YEAR ',I2,'    YEAR ',I2,'    YEAR ',I2,'    YEAR ',I2,
     6        '    YEAR ',I2,//,3X,
     7        'STUDENT SALARIES            ',5(1X,F9.0),/,3X,
     8        'STUDENT TRAVEL              ',5(1X,F9.0),/,3X,
     9        'INSTRUCTOR SALARY           ',5(1X,F9.0),/,3X,
     A        'ADMINISTRATIVE SALARY       ',5(1X,F9.0),/,3X,
     B        'EQUIPMENT                   ',5(1X,F9.0),/,3X,
     C        'FACILITIES                  ',5(1X,F9.0),/,3X,
     D        'INSTRUCTIONAL MATERIALS     ',5(1X,F9.0),/,3X,
     E          'SUPPLIES                     ',5(1X,F9.0),/,3X,
     F          'MISCELLANEOUS                ',5(1X,F9.0),/,3X,
     G          'TOTAL OPERATIONAL            ',5(1X,F9.0),/,3X,
     H          'COST PER GRADUATE            ',5(1X,F9.0),/,3X,
     I          'COST PER STUDENT INPUT       ',5(1X,F9.0),/,3X,
     J          'COST PER STUDENT POSITION    ',5(1X,F9.0))
C
C     WRITE MISSING STATEMENT NUMBERS
C
  960 IF (.NOT.RPT7) GO TO 990
      IF (K.LE.1) GO TO 990
      WRITE (6,5000)
      K = K-1
      WRITE (6,5001) ((MATA(I),MATB(I)),I=1,K)
 5000 FORMAT ('1','                                   MISSING DATA ANALYSIS')
 5001 FORMAT (1X,'MISSING DATA CODE = ',I6,4X,'CRIT CODE = ',I6)
C ***
C *** ETAM DESIGN MOD - THIS ROUTINE TEMPORARILY STORES THE RESULTS OF
C ***                   LOOP ONE - BASELINE RUN.
C ***                   RUN.
C ***
  990 IF (IND1.NE.0) GO TO 992
      DO 991 M=1,20
      ETAT01(M) = ETAGRD(M)
      ETAT02(M) = ETASTU(M)
      ETAT03(M) = ETAAOB(M)
      ETAT04(M) = ETAINS(M)
      ETAT05(M) = ETAADM(M)
      ETAT06(M) = ETAEQP(M)
      ETAT07(M) = ETAFAC(M)
      ETAT08(M) = ETASUP(M)
      ETAT09(M) = ETAMIS(M)
      ETAT10(M) = ETAINV(M)
      ETAT11(M) = ETAREC(M)
  991 IND1   = 1
      GO TO 10
C ***
C ***
C *** ETAM DESIGN MOD - THIS ROUTINE CALCULATES THE DIFFERENCE BETWEEN
C ***                   LOOP ONE - BASELINE RUN, AND LOOP TWO - BENEFIT
C ***
  992 DO 993 M=1,20
      ETAGRD(M) = ETAT01(M) - ETAGRD(M)
      ETASTU(M) = ETAT02(M) - ETASTU(M)
      ETAAOB(M) = ETAT03(M) - ETAAOB(M)
      ETAINS(M) = ETAT04(M) - ETAINS(M)
      ETAADM(M) = ETAT05(M) - ETAADM(M)
      ETAEQP(M) = ETAT06(M) - ETAEQP(M)
      ETAFAC(M) = ETAT07(M) - ETAFAC(M)
      ETASUP(M) = ETAT08(M) - ETASUP(M)
      ETAMIS(M) = ETAT09(M) - ETAMIS(M)
      ETAINV(M) = ETAT10(M) - ETAINV(M)
      ETAREC(M) = ETAT11(M) - ETAREC(M)
  993 IND1   = 0
      GO TO 995
C ***
C ***
C *** ETAM DESIGN MOD - THE FOLLOWING OUTPUT REPRESENTS THE COST MODEL
C ***                   RESULTS FOR COURSES WHICH WILL BE SUMMARIZED BY
C ***                   COURSE AND OUTPUTED TO THE CMR FILE FOR USE IN
C ***                   THE DECISION AND FINANCIAL ANALYSIS MODULES.
C ***                   IT REPRESENTS THE DIFFERENCE BETWEEN LOOP ONE
C ***                   WHICH CALCULATED BASELINE AND LOOP TWO WHICH
C ***                   HAD THE BENEFIT PATTERN APPLIED.
C ***
C
C     WRITE ETAM OUTPUT RESULTS
C
 1510 FORMAT ('1',28X,'ETAM OUTPUT FACTORS',//,32X,
     1          'YEAR ',I2,'   YEAR ',I2,'   YEAR ',I2,'   YEAR ',I2,
     2          '   YEAR ',I2,//,3X,
     3          'NO. OF GRADUATES RQD         ',5(1X,F9.0),/,3X,
     4          'NO. OF STUDENTS IN           ',5(1X,F9.0),/,3X,
     5          'AVERAGE ON BOARD             ',5(1X,F9.0),/,3X,
     6          'NO. OF INSTRUCTORS RQD       ',5(1X,F9.0),/,3X,
     7          'NO. OF ADMIN PERS RQD        ',5(1X,F9.0),/,3X,
     8          'NO. OF EQUIPMENTS RQD        ',5(1X,F9.0),/,3X,
     9          'NO. SQUARE FEET RQD          ',5(1X,F9.0),/,3X,
     A         'NO. SUPPLIES RQD              ',5(1X,F9.0),/,3X,
     B         'NO. MISCELLANEOUS RQD         ',5(1X,F9.0),//,3X,
     C         'ANNUAL INVESTMENT DOLLARS ',5(1X,F9.0),/,3X,
     D         'ANNUAL RECURRING CST/SAV  ',5(1X,F9.0))
  995 WRITE (6,1510) (DIGIT(I),I= 1, 5),
     1                (ETAGRD(I),I= 1, 5),(ETASTU(I),I= 1, 5),
     2                (ETAAOB(I),I= 1, 5),(ETAINS(I),I= 1, 5),
     3                (ETAADM(I),I= 1, 5),(ETAEQP(I),I= 1, 5),
     4                (ETAFAC(I),I= 1, 5),(ETASUP(I),I= 1, 5),
     5                (ETAMIS(I),I= 1, 5),(ETAINV(I),I= 1, 5),
     6                (ETAREC(I),I= 1, 5)
      WRITE (6,1510) (DIGIT(I),I= 6,10),
     1                (ETAGRD(I),I= 6,10),(ETASTU(I),I= 6,10),
     2                (ETAAOB(I),I= 6,10),(ETAINS(I),I= 6,10),
     3                (ETAADM(I),I= 6,10),(ETAEQP(I),I= 6,10),
     4                (ETAFAC(I),I= 6,10),(ETASUP(I),I= 6,10),
     5                (ETAMIS(I),I= 6,10),(ETAINV(I),I= 6,10),
     6                (ETAREC(I),I= 6,10)
      WRITE (6,1510) (DIGIT(I),I=11,15),
     1                (ETAGRD(I),I=11,15),(ETASTU(I),I=11,15),
     2                (ETAAOB(I),I=11,15),(ETAINS(I),I=11,15),
     3                (ETAADM(I),I=11,15),(ETAEQP(I),I=11,15),
     4                (ETAFAC(I),I=11,15),(ETASUP(I),I=11,15),
     5                (ETAMIS(I),I=11,15),(ETAINV(I),I=11,15),
     6                (ETAREC(I),I=11,15)
      WRITE (6,1510) (DIGIT(I),I=16,20),
     1                (ETAGRD(I),I=16,20),(ETASTU(I),I=16,20),
     2                (ETAAOB(I),I=16,20),(ETAINS(I),I=16,20),
     3                (ETAADM(I),I=16,20),(ETAEQP(I),I=16,20),
     4                (ETAFAC(I),I=16,20),(ETASUP(I),I=16,20),
     5                (ETAMIS(I),I=16,20),(ETAINV(I),I=16,20),
     6                (ETAREC(I),I=16,20)
C   ***
      IF (.NOT.LAST) GO TO 10
 1511 CONTINUE
 1512 STOP
      END
```

APPENDIX D

ETAM RANGE-OF-EFFECT PROGRAM DOCUMENTATION

62

SECTION D.1

INTRODUCTION

The purpose of this appendix is to provide documentation for those programs that have been implemented to date in the computerization of ETAM. Broadly, this includes programs in two general support areas - those modules used to construct the data bases used by the interactive ETAM system, and the programs which implement the Range-of-Effect search and reporting process.

Neither the ETAM system, nor the environment in which the current system is implemented is of a trivial nature. The key to interpretation of a majority of this appendix presumes at least a minimal working knowledge on the part of the reader with both basic ETAM and the programming requirements of the timesharing vendor.

ORGANIZATION

This appendix is organized into six major sections. Section D.1 illustrates the logic flow of the eighteen programs implemented in the current ETAM system. A separate coverage of each of the programs is made in Section D.2. Section D.3 documents the timesharing system executive routines that are common to more than one program. The details and formats of the online files required to support the ETAM system are covered in Section D.4. Input and control data used in the makeup of the online ETAM system are documented in Section D.5. In Section D.6 appear listings of the source cards for each of the programs in the current system. In summary, the contents of this appendix are:

Section D.1	Introduction
Section D.2	Program Descriptions
Section D.3	System Executive Routines
Section D.4	Online File Formats
Section D.5	Input and Control Data
Section D.6	Program Listings (Service Cards)

EQUIPMENT ENVIRONMENT

All ETAM programs are designed to run in an interactive mode on a remote time-shared computer. Timesharing services and support were obtained on a lease basis from National CSS, Inc., Norwalk, Connecticut. The National CSS VP/CSS* system is not documented within this publication. A basic knowledge of sign-on, sign-off, editing, and program execution are required of the ETAM User.

---

* Trademark of National CSS, Inc.

The ETAM User requires a standard remote terminal such as an IBM 2741, or a Teleterm 1030*. The terminal is connected to the timesharing vendor's computer via a dial-up telephone line and a modem. Further information on the distinct types of terminal equipments which can be employed in conjunction with the timesharing support should be obtained from National CSS, Inc.

As currently configured, the ETAM system utilizes thirty cylinders of online direct-access storage (disk storage) and all programs will function in a 256K byte partition of storage.

PROGRAM ENVIRONMENT

The implications associated with the following program documentation and VP/CSS executive sequences presupposes a working knowledge above that of the casual User with the basic services provided by the National CSS (NCSS) timesharing vendor. These services are formally documented by NCSS and are not reproduced within this manual. To assist those who are unfamiliar with the NCSS documentation, the following references are mentioned:

System Function	NCSS Reference
Typing Conventions, Login, Logout	A Guide to the National CSS Computer System for Application Product Users (NCSS Form 991) Section 1
System Interruptions	A Guide to the National CSS Computer System for Application Product Users (NCSS Form 991) Section 7
CSS COMMANDS SUMMARY	A Guide to the National CSS Computer System for Application Product Users (NCSS Form 991) Appendix B
CSS File Creation and Maintenance	A Guide to the National CSS Computer System for Application Product Users (NCSS Form 991) Section 3 VP/CSS EDit Command (NCSS Form 108)
VP/CSS EXECUTIVE Language	VP/CSS EXECUTIVE Language (EXEC) (NCSS Form 109-5)

---

* Trademark of Computer Devices, Inc.

ETAM SYSTEM INITIALIZATION

A total of eighteen programs are involved in the current ETAM system. In a broad sense, these can be divided into two distinct functional areas. Eleven programs are used in the generation and maintenance of the ETAM abbreviated data bases. The remaining seven programs implement the ETAM Range-of-Effect (ROE) search process.

Figure D.1-1 illustrates the basic initialization process that is required prior to the build of the abbreviated data bases. All files are indicated with a number; this "File Number" will be consistent throughout this appendix. The card files of Figure D.1-1 must be transmitted to the timesharing vendor's computer site and introduced to the online system via the terminal "Offline Read" process. Executive STARTUP has been designed to function with any or all of the card files indicated. This executive can be run multiple times, if required. All of the input data is sorted via executive STARTUP, and the Master Reference file (File Number 1) is set to zero by Program P18.

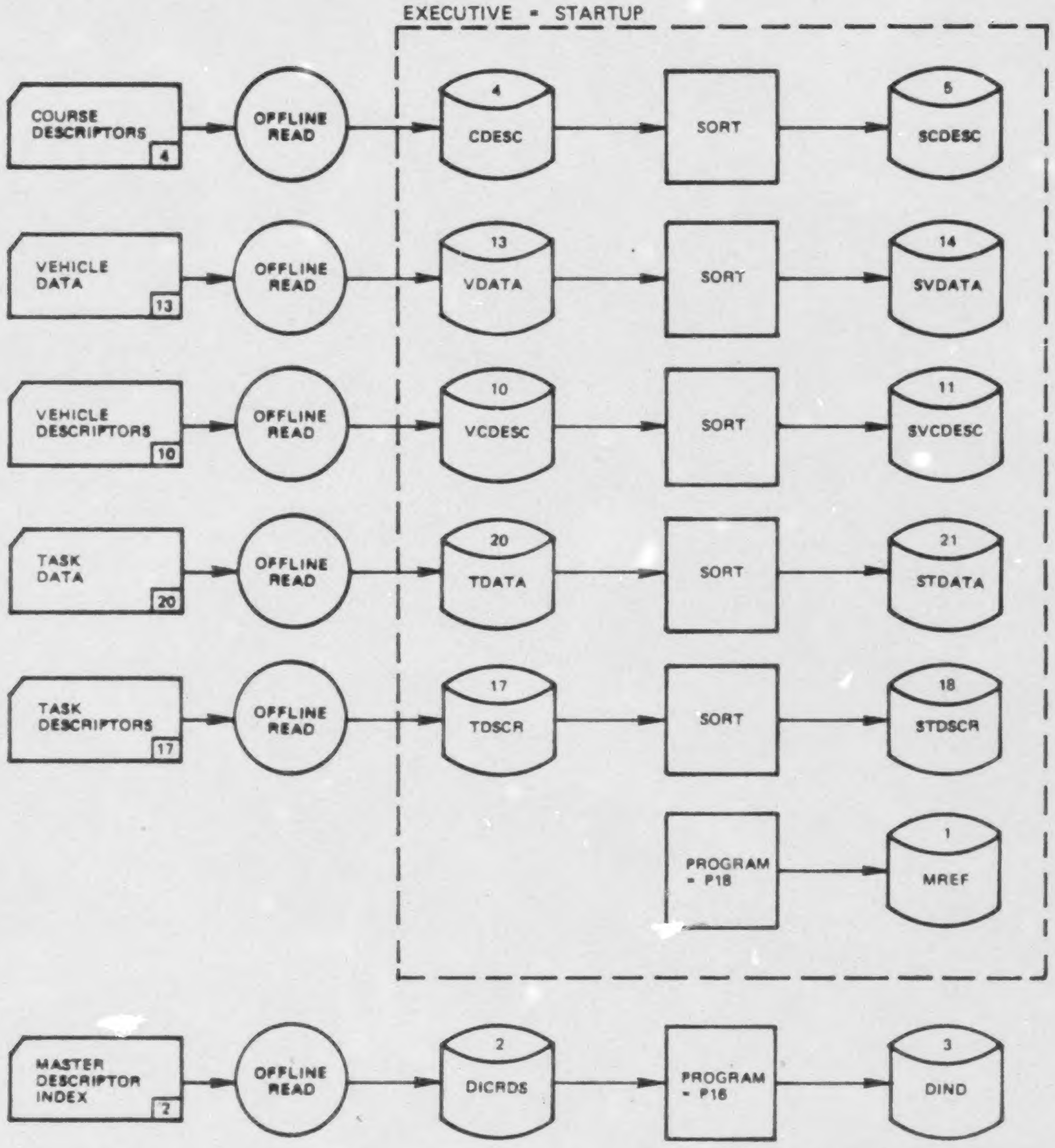
As indicated in Figure D.1-1, the Descriptor Master Index is created by running Program P16. All of the programs of the ETAM system are supported by a unique executive. The executive for each program bears the name "RUN" followed by the unique program name. Thus, Program P16 is executed by an invocation of Executive RUNP16.

LOAD OF THE ABBREVIATED COURSE DATA BASE

Figure D.1-2 illustrates the load of the four files of the Abbreviated Course data base. Each of the three data bases of the ETAM system (Course, Vehicle, and Task) is constructed by byilding a basic data file, a file directory, and a descriptor file. For the Course entity, two directories are required.

Program P14 uses the CIN (Course Identification Number) of the Course descriptor input card file as a selection criteria for obtaining Course data from the NITRAS tape. In the current implementation, descriptors have been assigned to 121 unique CIN's. The NITRAS tape contains data relative to more than 4000 Courses. The following load counts were extracted from the output of Program P14:

```
  132 : Total Descriptor Cards Read
  121 : Total Unique CIN's on Descriptor Cards
   50 : Total Number CIN's with No Data Match in NITRAS
  111 : Total Number Course Records extracted from NITRAS
4,127 : Total Number Course Records on NITRAS Tape
```

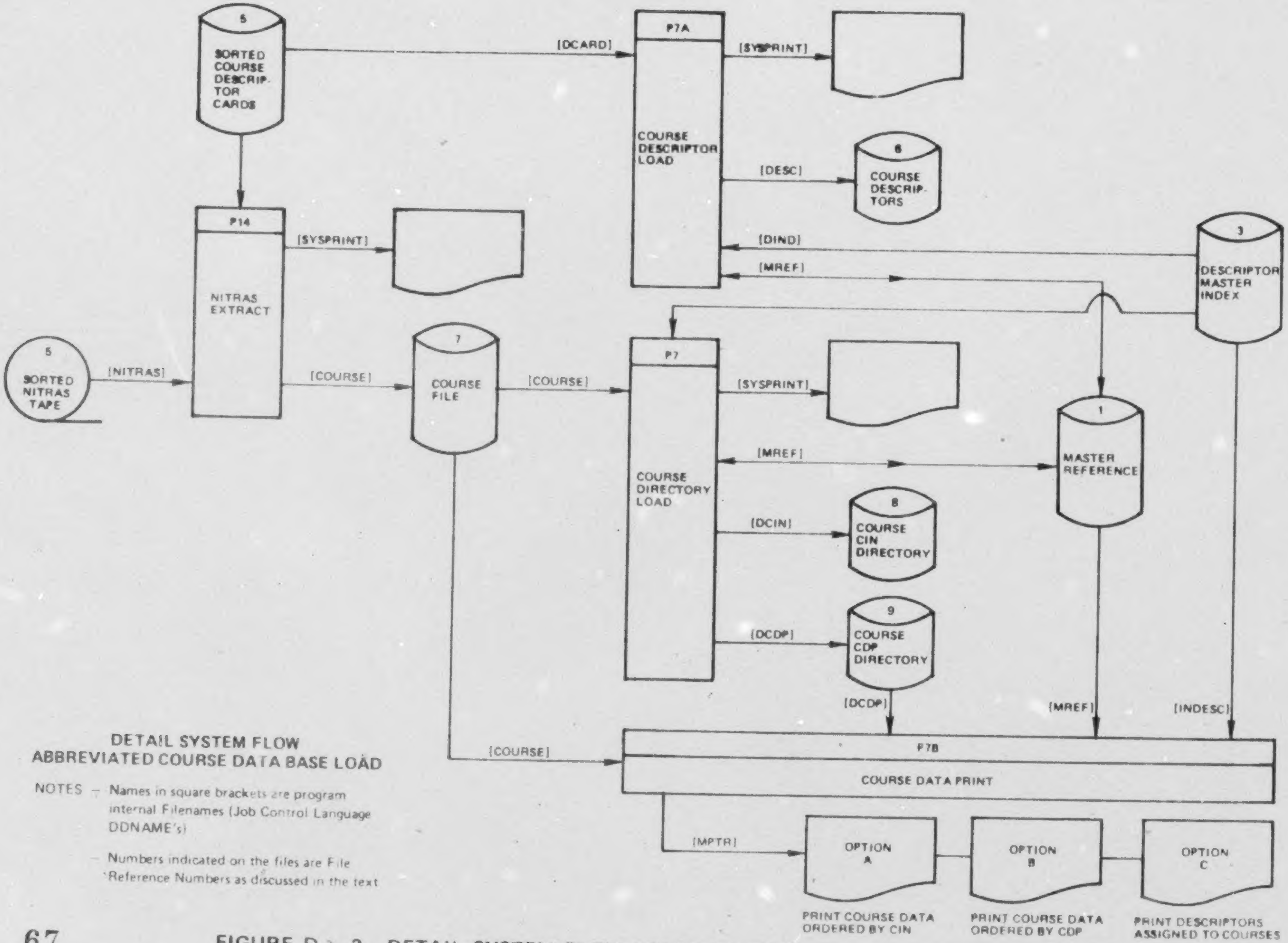

EXECUTIVE = STARTUP
COURSE DESCRIPTORS 4
OFFLINE READ
4 CDESC
SORT
5 SCDESC
VEHICLE DATA 13
OFFLINE READ
13 VDATA
SORT
14 SVDATA
VEHICLE DESCRIPTORS 10
OFFLINE READ
10 VCDESC
SORT
11 SVCDESC
TASK DATA 20
OFFLINE READ
20 TDATA
SORT
21 STDATA
TASK DESCRIPTORS 17
OFFLINE READ
17 TDSCR
SORT
18 STDSCR
PROGRAM = P18
1 MREF
MASTER DESCRIPTOR INDEX 2
OFFLINE READ
2 DICRDS
PROGRAM = P16
3 DIND

INITILIZATION (STARTUP) OF ETAM SYSTEM DATA SETS PRIOR TO CONSTRUCTION OF THE ABBREVIATED DATA BASES

All Files shown have filename = ETAM

Names shown above are filetype; e.g., the sorted course descriptor output of Executive STARTUP is called ETAM SCDESC

FIGURE D.1–1. INITIALIZATION (STARTUP) OF ETAM SYSTEM DATA SETS PRIOR TO CONSTRUCTION OF THE ABBREVIATED DATA BASES


5
SORTED COURSE DESCRIP-TOR CARDS
[DCARD]
P7A
COURSE DESCRIPTOR LOAD
[SYSPRINT]
[DESC]
6
COURSE DESCRIP-TORS
[DIND]
[MREF]
3
DESCRIPTOR MASTER INDEX
P14
NITRAS EXTRACT
[SYSPRINT]
5
SORTED NITRAS TAPE
[NITRAS]
[COURSE]
7
COURSE FILE
[COURSE]
P7
COURSE DIRECTORY LOAD
[SYSPRINT]
[MREF]
1
MASTER REFERENCE
[DCIN]
8
COURSE CIN DIRECTORY
[DCDP]
9
COURSE CDP DIRECTORY
[DCDP]
[MREF]
[INDESC]
[COURSE]
P7B
COURSE DATA PRINT
[MPTR]
OPTION A
OPTION B
OPTION C
PRINT COURSE DATA ORDERED BY CIN
PRINT COURSE DATA ORDERED BY CDP
PRINT DESCRIPTORS ASSIGNED TO COURSES

DETAIL SYSTEM FLOW
ABBREVIATED COURSE DATA BASE LOAD

NOTES – Names in square brackets are program internal Filenames (Job Control Language DDNAME's)

– Numbers indicated on the files are File Reference Numbers as discussed in the text

FIGURE D.1–2. DETAIL SYSTEM FLOW ABBREVIATED COURSE DATA BASE LOAI

The two Course directories are constructed by Program P7. The Abbreviated Course data file is used as the source. A directory ordered by Course CIN and a directory ordered by Course CDP are required.

The Course descriptor card input is used by Program P7A to construct the Course descriptor file (File Number 6). This file will be the reference to be used in the ETAM Range-of-Effect search process.

Each of the programs of the data base build processes introduce counts into the Master Reference file (File Number 1). This file will be used by all programs which reference the abbreviated data bases. These files will be accessed randomly and the counts provided by the Master Reference file will be used as program internal selfchecks to prevent input/output errors.

The final program of the Course data base build process is the Course Print Program, Program P7B. This program offers interactive options to request print of the Course data base contents.

Program P20 is not shown on the diagrams of this appendix. This program's function is to print the entities of a selected data base, followed by all descriptors applicable to each entity. This program provides an interactive option to allow print of Course entities plus descriptor data.

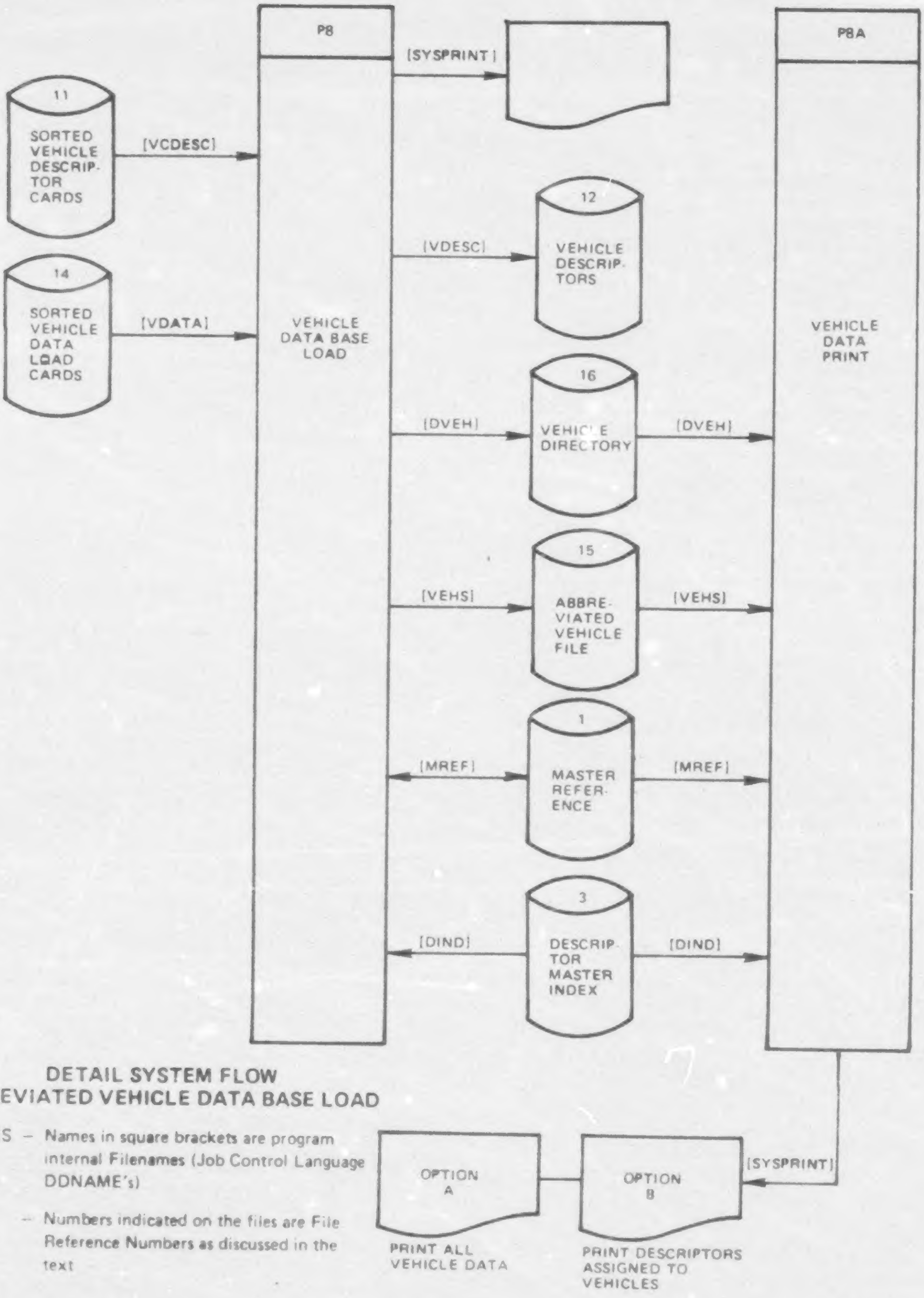
ABBREVIATED VEHICLE DATA BASE LOAD

The load of the three files (data, directory, and descriptor) of the Vehicle data base is illustrated in Figure D.1-3. Both the data file and the descriptor file are loaded from card input as transmitted to the timesharing computer system. At present, data coverage for 63 unique Vehicle stock numbers has been provided. Record counts relative to the Vehicle data files are written into the Master Reference file by Program P9.

Printer dumps of the Abbreviated Vehicle data base contents are the output of Program P8A. Options (interactive) supporting data file print and print of the general Vehicle type descriptors are available. Program P20 (not shown on Figure D.1-3) can be used to produce a listing of all Vehicle entities and the specific descriptors applied to each.

ABBREVIATED TASKS DATA BASE LOAD

The build of the Abbreviated Tasks data base is shown in D.1-4. This is the largest of the three ETAM data bases. The current system supports 670 distinct Tasks. Both the Tasks data file and descriptor file are loaded from card input. Due to the bulk of the output for the Vehicle entity, Program P9 allows selection of the printer output device. The

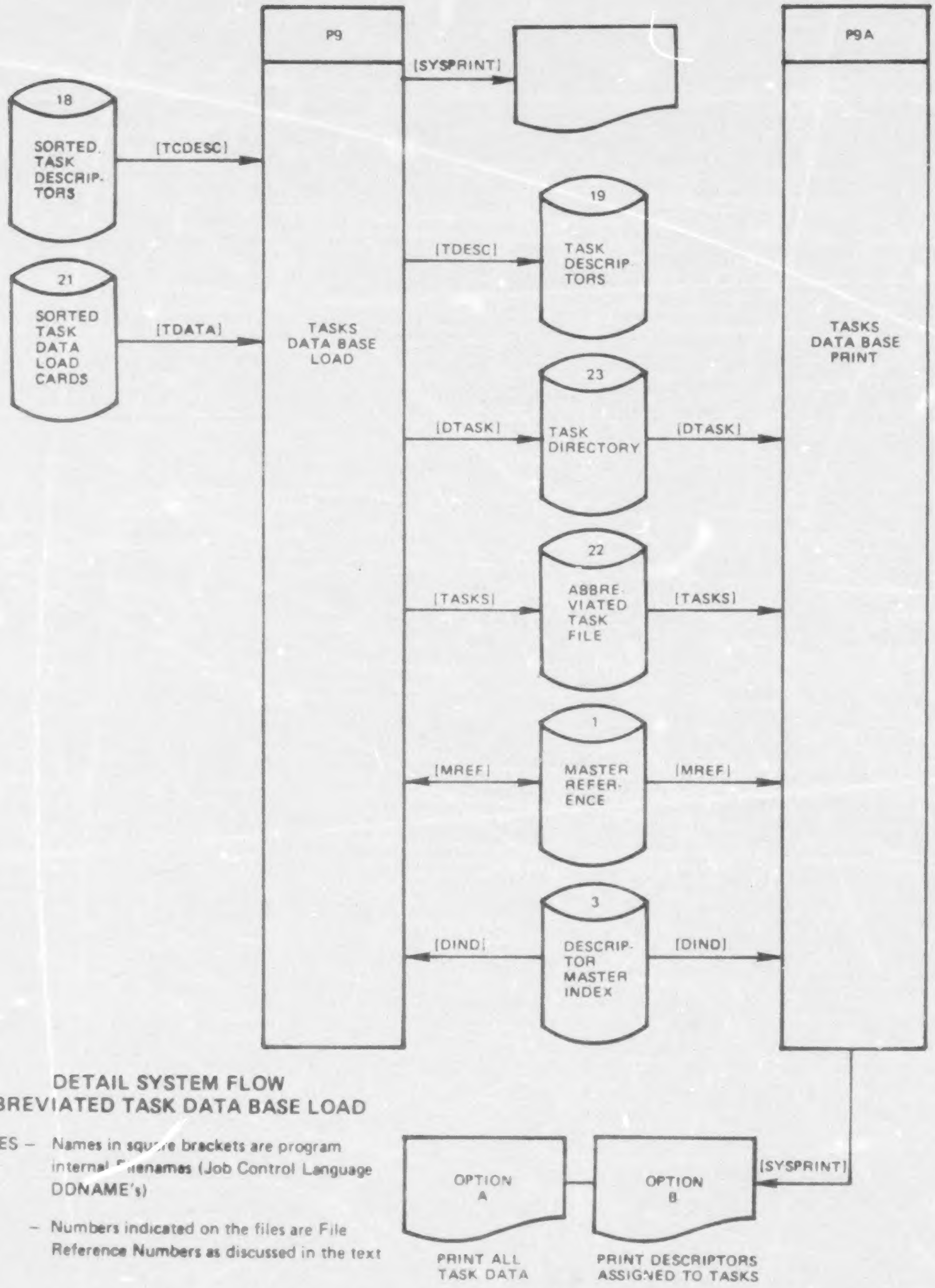

P8
[SYSPRINT]
P8A
11
SORTED VEHICLE DESCRIP-TOR CARDS
[VCDESC]
12
[VDESC]
VEHICLE DESCRIP-TORS
14
SORTED VEHICLE DATA LOAD CARDS
[VDATA]
VEHICLE DATA BASE LOAD
VEHICLE DATA PRINT
16
[DVEH]
VEHICLE DIRECTORY
[DVEH]
15
[VEHS]
ABBRE-VIATED VEHICLE FILE
[VEHS]
1
[MREF]
MASTER REFER-ENCE
[MREF]
3
[DIND]
DESCRIP-TOR MASTER INDEX
[DIND]
OPTION A
OPTION B
[SYSPRINT]
PRINT ALL VEHICLE DATA
PRINT DESCRIPTORS ASSIGNED TO VEHICLES

DETAIL SYSTEM FLOW
ABBREVIATED VEHICLE DATA BASE LOAD

NOTES – Names in square brackets are program internal Filenames (Job Control Language DDNAME's)

– Numbers indicated on the files are File Reference Numbers as discussed in the text

FIGURE D.1–3. DETAIL SYSTEM FLOW ABBREVIATED VEHICLE DATA BASE LOAD


P9
[SYSPRINT]
P9A
18
SORTED TASK DESCRIPTORS
[TCDESC]
21
SORTED TASK DATA LOAD CARDS
[TDATA]
TASKS DATA BASE LOAD
[TDESC]
19
TASK DESCRIPTORS
[DTASK]
23
TASK DIRECTORY
[DTASK]
[TASKS]
22
ABBREVIATED TASK FILE
[TASKS]
[MREF]
1
MASTER REFERENCE
[MREF]
[DIND]
3
DESCRIPTOR MASTER INDEX
[DIND]
TASKS DATA BASE PRINT
[SYSPRINT]
OPTION A
OPTION B
PRINT ALL TASK DATA
PRINT DESCRIPTORS ASSIGNED TO TASKS

DETAIL SYSTEM FLOW
ABBREVIATED TASK DATA BASE LOAD

NOTES – Names in square brackets are program internal filenames (Job Control Language DDNAME's)

– Numbers indicated on the files are File Reference Numbers as discussed in the text

FIGURE D.1–4. DETAIL SYSTEM FLOW ABBREVIATED TASK DATA BASE LOAD

online terminal, offline printer, or both may be chosen interactively by the User. Program P9 inserts record counts from the Vehicle data base files into the Master Reference file (File Number 1).

Functioning in a manner parallel to that of Program P8A for Vehicle data, Program P9A will produce printer dumps of the contents of the Tasks data base. For a listing of Task entity and all related descriptors, Program P20 must be used.

ESTABLISHING A NEW ETAM PROJECT

All of the files generated thus far have been ETAM "System" files which are of equal utility to all ETAM Projects. The files covered from this point forward are unique to each unique ETAM project.

The first step in the introduction of a new ETAM Project to the ETAM System is accomplished by running Program P17. This is illustrated on Figure D.1-5. The four Project Data Base files of interest in the Range-of-Effect search are the ID, REC, REV, and REJ files (File Numbers 26, 27, 28, and 29). These files contain an identification of the specific project and indicate those descriptors that are to be used in the forthcoming ROE search. The other Project Data Base files that can be created via Program P17 are of textual interest only at this point.

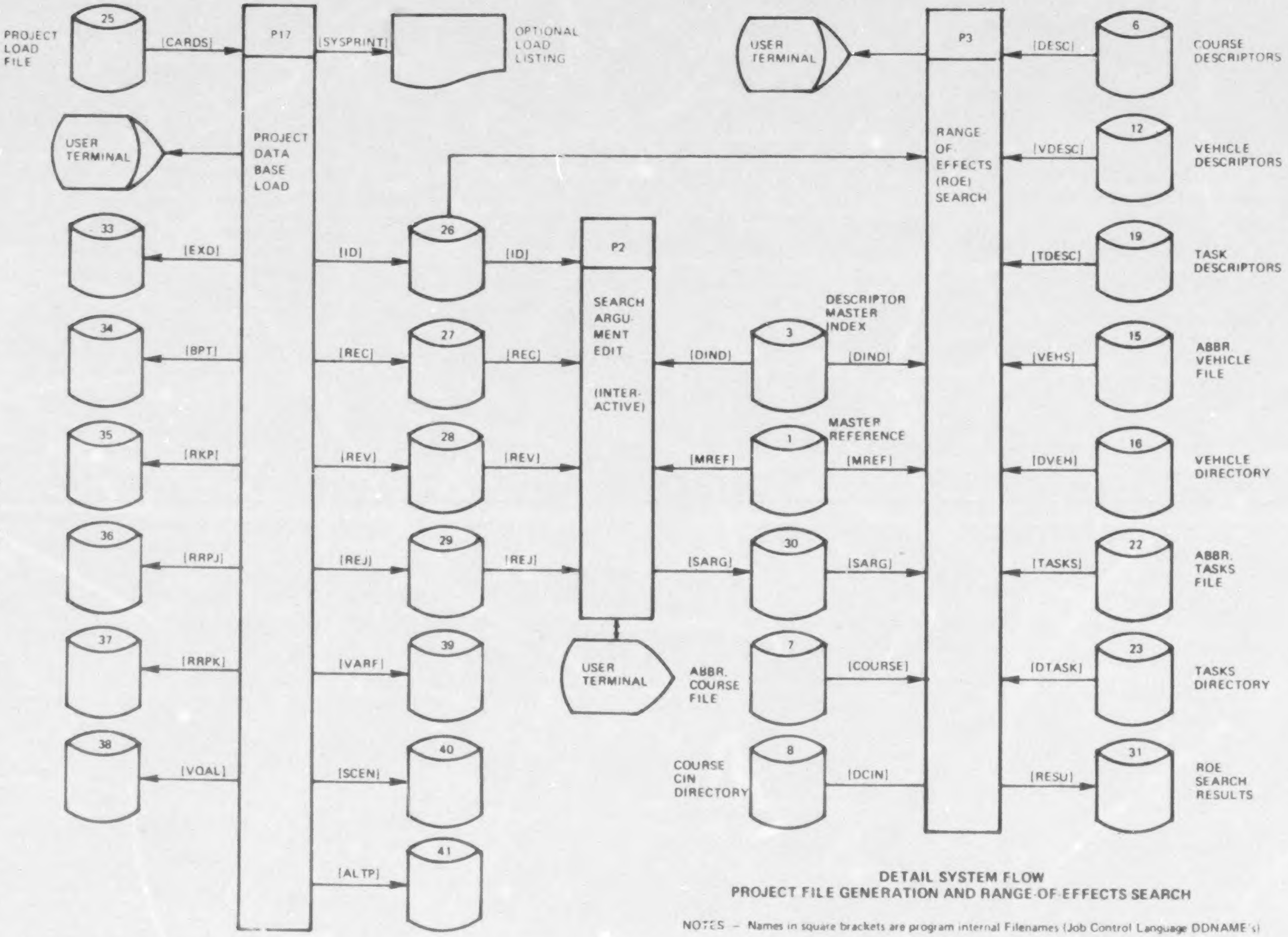
Program P1 can be used to print any/all of the Project Data Base file contents for a specific project.

EXECUTION OF THE RANGE-OF-EFFECT SEARCH

The Search Argument Edit Program P2 accomplishes the first step of the Range-of-Effect search process. The search arguments for the entities Courses, Vehicles, and Tasks are located in files REC, REV, and REJ, respectively. The User may peruse those descriptors already in the system and delete or add as is necessary. The final action of Program P2 is to copy the edited results into the search argument file (File Number 30).

File SARG (File Number 30) is generated as a result of running Program P2. As shown in Figure D.1-5, this is the only method of generating the SARG file. Since the SARG file is the prime input to the Search Program P3, it is required that Program P2 be executed prior to executing Program P3.

Program P3 accomplishes the actual search. No extensive output is directed to the terminal by Program P3. The search results are accumulated into the RESU file (File Number 31). The User is informed by terminal output as to the size (number of records selected as a result of the search) of the results file RESU. If, due to the number of records selected in the search,


PROJECT LOAD FILE
25
[CARDS]
P17
PROJECT DATA BASE LOAD
[SYSPRINT]
OPTIONAL LOAD LISTING
USER TERMINAL
33
[EXD]
34
[BPT]
35
[RKP]
36
[RRPJ]
37
[RRPK]
38
[VQAL]
[ID]
26
[REC]
27
[REV]
28
[REJ]
29
[VARF]
39
[SCEN]
40
[ALTP]
41
P2
SEARCH ARGUMENT EDIT
(INTERACTIVE)
USER TERMINAL
DESCRIPTOR MASTER INDEX
3
[DIND]
MASTER REFERENCE
1
[MREF]
30
[SARG]
ABBR. COURSE FILE
7
[COURSE]
COURSE CIN DIRECTORY
8
[DCIN]
USER TERMINAL
P3
RANGE OF EFFECTS (ROE) SEARCH
[DESC]
6
COURSE DESCRIPTORS
[VDESC]
12
VEHICLE DESCRIPTORS
[TDESC]
19
TASK DESCRIPTORS
[VEHS]
15
ABBR VEHICLE FILE
[DVEH]
16
VEHICLE DIRECTORY
[TASKS]
22
ABBR. TASKS FILE
[DTASK]
23
TASKS DIRECTORY
[RESU]
31
ROE SEARCH RESULTS

DETAIL SYSTEM FLOW
PROJECT FILE GENERATION AND RANGE-OF-EFFECTS SEARCH

NOTES – Names in square brackets are program internal Filenames (Job Control Language DDNAME's)

– Numbers indicated on the files are File Reference Numbers as discussed in the text

FIGURE D.1–5. DETAIL SYSTEM FLOW: PROJECT FILE GENERATION AND RANGE-OF-EFFECT SEARCH

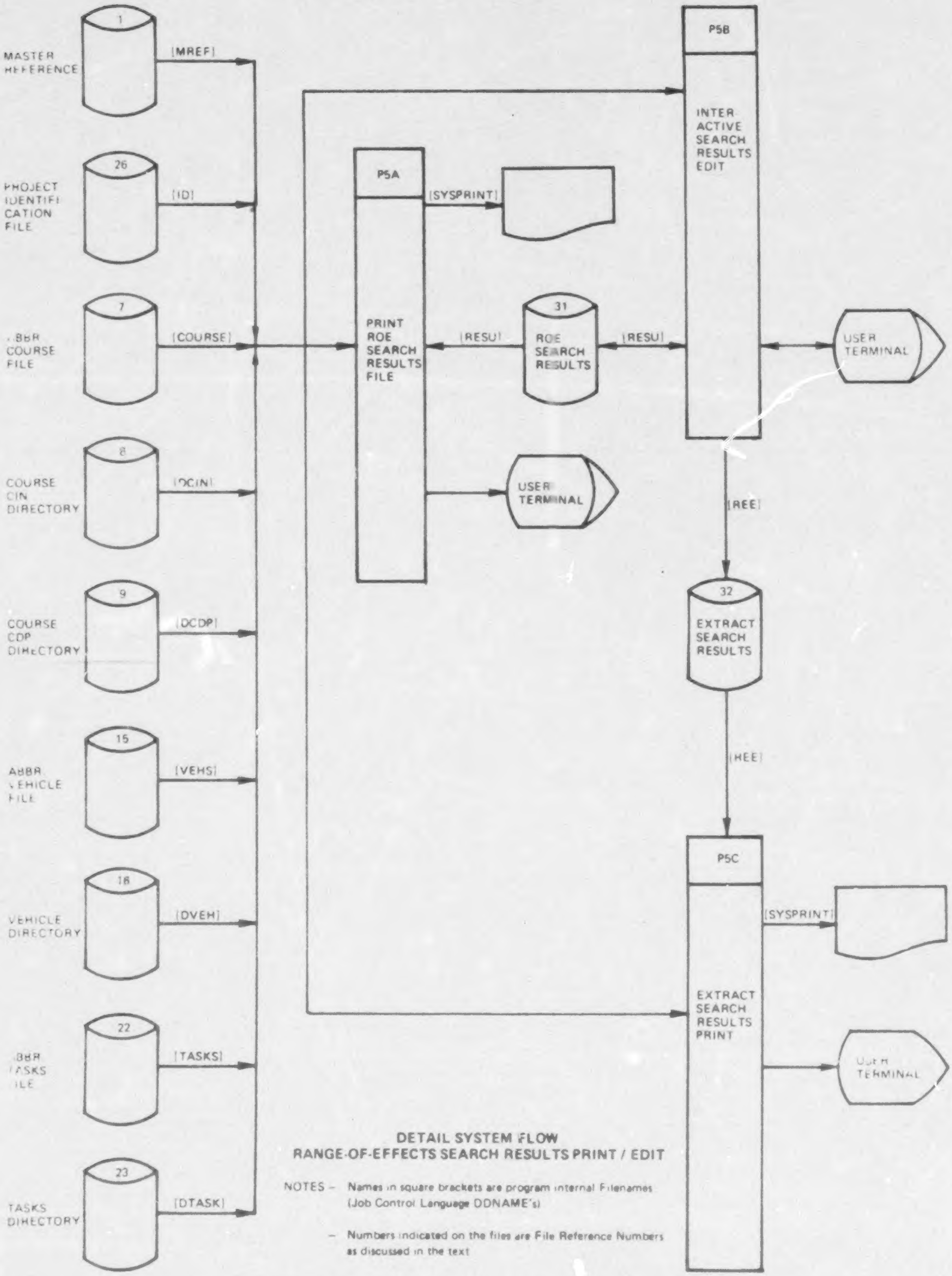
the User determines that an obvious error has been made in the selection of search arguments, the argument set may be edited by re-running Program P2. A subsequent run of the search Program P3 will create a new results file.

PROCESSING OF SEARCH RESULTS

Figure D.1-6 indicates the logic flow of the three programs that are involved in the processing of the results of a Range-of-Effect search. Program P5A is used to print the results of a search; the results file RESU is used as the input to Program P5A. This program also numbers each of the records as they are printed. This is vital since the results edit program (Program P5B) requires that record numbers be entered as a part of the interactive commands.

Program P5B allows the User to add or delete records from the search results. Upon termination, Program P5B copies all original and added results into the extract search results file REE (File Number 32).

Program P5C prints the contents of the extract results file in a finished format. This listing is the conclusion of the Range-of-Effect search process.


MASTER REFERENCE
1
[MREF]
PROJECT IDENTIFICATION FILE
26
[ID]
ABBR COURSE FILE
7
[COURSE]
COURSE CIN DIRECTORY
8
[DCIN]
COURSE CDP DIRECTORY
9
[DCDP]
ABBR VEHICLE FILE
15
[VEHS]
VEHICLE DIRECTORY
16
[DVEH]
ABBR TASKS FILE
22
[TASKS]
TASKS DIRECTORY
23
[DTASK]
P5A
PRINT ROE SEARCH RESULTS FILE
[SYSPRINT]
USER TERMINAL
31
ROE SEARCH RESULTS
[RESU]
[RESU]
P5B
INTER ACTIVE SEARCH RESULTS EDIT
USER TERMINAL
[REE]
32
EXTRACT SEARCH RESULTS
[REE]
P5C
EXTRACT SEARCH RESULTS PRINT
[SYSPRINT]
USER TERMINAL

DETAIL SYSTEM FLOW
RANGE-OF-EFFECTS SEARCH RESULTS PRINT / EDIT

NOTES – Names in square brackets are program internal Filenames (Job Control Language DDNAME's)

– Numbers indicated on the files are File Reference Numbers as discussed in the text

FIGURE D.1–6. DETAIL SYSTEM FLOW RANGE-OF-EFFECT SEARCH RESULTS PRINT/EDIT

SECTION D.2

PROGRAM DOCUMENTATION

In this Section are documentation packages covering each of the eighteen programs of the current ETAM system implementation. Specific information relative to each program is provided in the following general areas:

(1) Executive support required

(2) Itemization of all data files processed (input and output), their internal/external names and data formats

(3) Discussion of program function

(4) Listings of any executive routine(s) associated only with the program being covered

(5) Specific User instructions as to the use of program features

(6) Error messages generated and their meaning

(7) Sample output resulting from program execution

PROGRAM NAME : P1

ENTRY POINT : P1

RUN EXECUTIVES(S) : RUNP1, NONAME, IFNAME, JCL, JCL2

USER OUTPUT VIA : Terminal, Offline Printer, or Both

USER PROMPTING : Internal to Program: (1) File(s) to be Printed
(2) Output Device(s) Selection

FUNCTION : Prints Contents of Any/All Project Files

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
SYSPRINT	OUTPUT	N/A : THIS IS THE OFFLINE PRINTER			FA, 121, 121
ID	INPUT	26	DASD	Projectname/ID	F, 80, 80
EXD	INPUT	33	DASD	Projectname/EXD	F, 80, 80
BPT	INPUT	34	DASD	Projectname/BPT	F, 80, 80
RKP	INPUT	35	DASD	Projectname/RKP	F, 80, 80
RRPJ	INPUT	36	DASD	Projectname/RRPJ	F, 80, 80
RRPK	INPUT	37	DASD	Projectname/RRPK	F, 80, 80
VQAL	INPUT	38	DASD	Projectname/VQAL	F, 80, 80
VARF	INPUT	39	DASD	Projectname/VARF	F, 80, 80
SCEN	INPUT	40	DASD	Projectname/SCEN	F, 80, 80
ALTP	INPUT	41	DASD	Projectname/ALTP	F, 80, 80
REC	INPUT	27	DASD	Projectname/REC	F, 80, 80
REV	INPUT	28	DASD	Projectname/REV	F, 80, 80
REJ	INPUT	29	DASD	Projectname/REJ	F, 80, 80
RESU	INPUT	31	DASD	Projectname/RESU	FB, 16, 800
REE	INPUT	32	DASD	Projectname/REE	FB, 16, 800
MREF	INPUT	1	DASD	ETAM/MREF	

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
DIND	DIRECT	3	DASD	ETAM/DIND	
COURSE	DIRECT	7	DASD	ETAM/COURSE	
DCIN	INPUT	8	DASD	ETAM/DCIN	
DCDP	INPUT	9	DASD	ETAM/DCDP	
VEHS	DIRECT	15	DASD	ETAM/VEHS	
DVEH	INPUT	16	DASD	ETAM/DVEH	
TASKS	DIRECT	22	DASD	ETAM/TASKS	
DTASK	INPUT	23	DASD	ETAM/DTASK	

Discussion

This program has as its function the printing of all files of the ETAM system called "Project Files". Program P1 has been designed to allow the interactive selection of any or all of the files to be printed.

Listing of EXECUTIVE =RUNP1

```
          P RUNP1 EXEC

     &TYPE OFF
     &COMMENT INTERACTIVE PROJECT FILE PRINT PROGRAM.
     &COMMENT SINGLE ENTRY PARM REQUIRED - PROJECT NAME.
     &COMMENT EXEC = DEVICE WILL PROMPT FOR OUTPUT DEVICE.
     &IF &INDEX EQ 1 &GOTO -STP1
EXEC NONAME RUNP1
-STP1 &ALPHA3 = &1
EXEC IFNAME
EXEC JCL
EXEC JCL2
FILEDEF SYSPRINT PTR RE FA LR 121 BL 121
LOAD P1 (CLEAR LIBE) PLILIB
START (BRIEF)
FILEDEF * CLEAR
     &EXIT
```

Program P1 Specific User Instructions

This program processes only those files associated with a unique ETAM project within a single run. Accordingly, the name of the project must be provided by the User when invoking the RUNP1 executive. On the sample runs to follow, this is demonstrated using the ETAM project name "PN123".

Upon entry and after each processing command has been completed, the User will be prompted for entry as follows:

```
** ENTER FILETYPE FOR OUTPUT, MENU, OR QUIT **
```

A "MENU" selection will cause the printing of a list of filetypes and titles for output. Following the presentation, a return is made to the prompt message as above. The "QUIT" option is self-explanatory. If a valid filetype is entered, the system will then prompt for the desired output device, as follows:

```
OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
ENTER TERM, PRTR, BOTH, OR QUIT
```

The response options are self-explanatory to this prompt.

If, instead of a specific filetype, a response of "ALL" is made to the first general prompt, above, all filetypes associated with the current project will be output. Following the "ALL" selection, the device selected for output will be used for all output. After processing an "ALL" option, return is made to the first general prompt as usual.

Program P1 Error Messages

If the response to a specific prompt is not decodeable, the following error message is output:

```
** LAST ENTRY UNRECOGNIZED - TRY AGAIN **
```

The above message will be followed by a return to the general prompt message that preceeded it.

In response to the prompt for output device, an unrecognized input will result in the following:

```
UNABLE TO DETERMINE ENTRY TYPE - TRY AGAIN
```

(followed by a return to allow another device selection)

In the current ETAM implementation, the following filetypes have not been fully specified:

Filetype	Project File
EXC	Course Extract Data Base
EXJ	Tasks (Jobtasks) Extract Data Base
EXV	Vehicle Extract Data Base
CMR	Cost Model Results
TRER	Tree Results
FINR	Financial Results

An attempt to print one of these filetypes will result in the following message:

```
** FILETYPE xxxx IS NOT YET SERVICEABLE, TRY AGAIN **
```

where the name of the filetype of concern replaces "xxxx" in the message.

Program P1 Sample Run Output

The following illustrates a run of Program P1 servicing the Project Files of Project "PN123".

```
          RUNP1 PN123

PROJECT NAMED PN123 ALREADY EXISTS,
IS THIS CORRECT? (RESPOND YES OR QUIT)

YES
$$$$$
GENERAL-PURPOSE PROJECT FILE PRINT
PROGRAM IS NOW STARTING...

::: ENTER FILETYPE FOR OUTPUT, MENU, OR QUIT :::
_MENU

PROJECT FILETYPES FOR OUTPUT ARE :

     ID - PROJECT DESCRIPTION
     EXD - EXTRACT DEFAULTS
     BPT - BENEFIT PATTERN
     RKP - RISK PROFILE
     RRPJ - RISK REDUCTION PROJECTS
     RRPK - RISK REDUCTION PACKAGES
     VQAL - VARIABLES QUALIFICATION
     VARF - VARIABLE REFERENCES
     SCEN - SCENARIOS
     ALTP - ALTERNATE PROJECTS
     REC - COURSES ROE SEARCH ARGUMENTS
     REV - VEHICLES ROE SEARCH ARGUMENTS
     REJ - TASKS ROE SEARCH ARGUMENTS
     RESU - ROE TOTAL SEARCH RESULTS
     REE - ROE SEARCH RESULTS
     EXC - EXTRACT DB - COURSES
     EXJ - EXTRACT DB - JOBTASKS
     EXV - EXTRACT DB - VEHICLES
     CMR - COST MODEL RESULTS
     TRER - TREE RESULTS
     FINR - FINANCIAL RESULTS

** ENTER FILETYPE FOR OUTPUT, MENU, OR QUIT **
_REC

OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT
ENTER TERM, PRTR, BOTH, OR QUIT
_BOTH


     PAGE NUMBER :     1

LISTING OF PROJECT FILE CONTENTS
         PN123      / REC - COURSES ROE SEARCH ARGUMENTS


        30  OPERATIONS

        31  MAINTENANCE

        40  EQUIPMENT & OBJECTS USED: REAL

        64  PROCEDURE FOLLOWING

       -80  ORIENTATION, FAMILIARIZATION

       -81  TASK NOMENCLATURE, IDENTS, LOCATIONS, FACTS, RULES

        82  TASK FORMATS AT CONCEPTUAL LEVEL

        83  PROCEDURES AT VERBAL LEVEL ONLY

        84  TASK COMPONENTS WITH GUIDANCE

        85  ENTIRE JOB-TASK PROCEDURALLY AT BARELY ACCEPTABLE MASTERY

        86  HIGHLY PROFICIENT IN WORK CONTEXT

          A TOTAL OF   11 RECORDS ARE PRESENT
          WITHIN THIS FILE

        ** ENTER FILETYPE FOR OUTPUT, MENU, OR QUIT **
        _QUIT


        PROJECT-FILE PRINT PROGRAM IS NOW TERMINATING
```

PROGRAM NAME : P2

ENTRY POINT : P2

RUN EXECUTIVES(S) : RUNP2, NONAME, IFNAME, JCL, JCL2

USER OUTPUT VIA : Terminal Only

USER PROMPTING : Internal Program Only

FUNCTION : Interactively Modify/Examine the Contents of the ROE Search Argument Files REC, REV, and REJ (File Reference Numbers 27, 28, and 29)

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
MREF	INPUT	1	DASD	ETAM/MREF	F, 80, 80
DIND	DIRECT	3	DASD	ETAM/DIND	F, 80, 80
SARG	OUTPUT	30	DASD	Projectname/SARG	F, 240, 240
ID	INPUT	26	DASD	Projectname/ID	F, 80, 80
REC	INOUT/ OUTPUT	27	DASD	Projectname/REC	F, 80, 80
REV	INOUT/ OUTPUT	28	DASD	Projectname/REV	F, 80, 80
REJ	INOUT/ OUTPUT	29	DASD	Projectname/REJ	F, 80, 80

Discussion

This program serves a twofold purpose:

For the User : This program allows interactive printing, adding, and deleting of descriptors to be used as search arguments in future Range-of-Effects (ROE) searches.

For the System : Program P2 creates a search argument file (File Number 30) using the descriptors supplied via Project Files REC, REV, and REJ. The ROE Search Program (Program P3) cannot be run successfully until this file has been created.

NOTE - This implies that Program P2 must be run <u>at least one time</u> prior to running Program P3.

<u>Listing of EXECUTIVE = RUNP2</u>

```
    &TYPE OFF
    &COMMENT INTERACTIVE EDIT OF SEARCH ARGUMENTS FROM PROJECT FILES
    &COMMENT REC, REV, REJ TO THE SEARCH ARGUMENT PF = SARG.
    &COMMENT SINGLE ENTRY PARM REQUIRED IS PROJECT NAME.
    &COMMENT OUTPUT ONLY TO TERMINAL IN THIS SEQUENCE.
    &IF &INDEX EQ 1 &GOTO -STP1
EXEC NONAME RUNP2
-STP1 &ALPHA3 = &1
EXEC IFNAME
EXEC JCL
EXEC JCL2
LOAD P2 (CLEAR LIBE) PLILIB
START (BRIEF)
FILEDEF * CLEAR
    &EXIT
```

<u>Program P3 Specific User Instructions</u>

Program P2 processes only those files associated with a unique ETAM project during a single run. It is mandatory that the name of the specific project be provided by the User when invoking the RUNP2 executive. On the sample runs to follow, this is illustrated using the ETAM project name "PN123".

This program has been designed to work in an interactive manner; all User input/output is through the online terminal.

After program initialization, the following message will appear at the terminal:

```
    ** MAKE INITIAL ENTRY SELECT
```

The anticipated response to this prompt is the word "SELECT" followed by one of the options "COURSES", "VEHICLES", or "TASKS". This allows the User to select one of the three ETAM search entities for consideration. An illegal entry at this point (usually mispelling) will be flagged by the message:

```
** SELECTION ERROR - TRY AGAIN

(followed by the initial entry prompt)
```

A successful "SELECT" entry results in the following output:

```
** XXXXXXXX  SEARCH DESCRIPTORS SELECTED

**                YY DESCRIPTORS NOW SELECTED
```

where "XXXXXXXX" will indicate the entity type selected, and "YY" indicates the total number of descriptor search arguments presently selected for this Project. The above is then followed by:

```
** MAKE NEXT ACTION REQUEST
```

This message is referred to in the following as the "general prompt". It follows all User entry to allow further entry. The User may, at this point, make one of the following entries:

```
SELECT   (followed by)   COURSES  or  VEHICLES  or  TASKS

ADD      number

DELETE   number

PRINT

FILE

QUIT
```

The "SELECT" entry functions in the same manner as discussed previously. The "ADD" and "DELETE" entries allow the User to modify the contents of the REC, REV, and REJ files. The "number" fields represent the descriptor numbers to be added or deleted. Once an entity type (Courses, Vehicles, or Tasks) has been selected, the "ADD", "DELETE", and "PRINT" verbs apply only to descriptors of that entity type until a new type is selected.

Selection of the "QUIT" option terminates Program P2 without permanent modification to any of the search designator (argument) files.

The "FILE" option terminates Program P2; all files to be modified are altered permanently in accordance with the interactive entry commands processed during Program P2 operation.

Program P2 Error Messages

During Program P2 initialization, the contents of the REC, REV, and REJ Project Files are input. These files were loaded previously by a run of Program P17. P17 merely loaded the files with no error diagnosis on the contents. Program P2 accomplishes error detection during their load. In the event of an invalid descriptor, the descriptor is removed from the file of its origin and the following message results:

```
** ILLEGAL DESCRIPTOR -  XXXX  -  DETECTED IN YYYYYYYY FILE **
   DESCRIPTOR WILL BE IGNORED AND REMOVED FROM THE YYYYYYYY FILE
```

In the above message, "XXXX" will indicate the invalid descriptor number and "YYYYYYYY" the entity type (Course, Vehicle, or Tasks).

The following general purpose message results for any entry error following the general prompt message:

```
** UNIDENTIFIED ACTION REQUEST  -  TR[illegible]AIN
```

An illegal descriptor number will be signified by:

```
** KEYBOARD ENTRY  - XXXX -  CONTAINED ILLEGAL NUMERIC CHARACTER
```

Selection of a descriptor number that does not apply to the selected entity type results in:

```
** XXXX IS NOT A LEGAL DESCRIPTOR FOR YYYYYYYY TYPE
```

The following self-explanatory message results from an illegal deletion request:

```
** DESIGNATOR NUMBER XXXX HAS NOT BEEN SELECTED;
   IT CANNOT BE DELETED
```

A general-purpose error message can result during Program P2 operation:

```
** ERROR NUMBER  XX  HAS OCCURED **
```

(followed by a general data-output line)

In the above, the error number "XX" has the following meaning:

01, - Error occured during "PRINT" option processing.
02

Direct access indexing into the Master Descriptor Index file (File Number 3) failed. The second error line presents the indices of concern.

03 - While processing an "ADD" entry, an indexing problem arose into the same file as error number 1, above.

04 - This is an internal program indexing problem encountered while attempting rewrite of the descriptor search argument files ("FILE" option processing).

Program P2 Sample Run Output

The following sample illustrates a complete run of Program P2. Only the "COURSES" type descriptors are referenced in this run.

```
          RUNP2 PN123

PROJECT NAMED PN123 ALREADY EXISTS,
IS THIS CORRECT? (RESPOND YES OR QUIT)

YES
$$$
          SEARCH DESCRIPTOR EDITOR RUN FOR PROJECT : PN123
3-D PROCEDURAL TRAINER


** MAKE INITIAL ENTRY SELECT
_SELECT COURSES

** COURSES  SEARCH DESCRIPTORS SELECTED

**        11 DESCRIPTORS NOW SELECTED

** MAKE NEXT ACTION REQUEST
_PRINT

** GROSS JOB CATEGORIES
30 OPERATIONS
31 MAINTENANCE

** OBJECTIVE TASK VARIABLES AS MANIFEST IN THE TRAINING
40 EQUIPMENT & OBJECTS USED: REAL

** TASK FUNCTIONS DOMINANT IN TRAINING
64 PROCEDURE FOLLOWING

** STAGE OF LEARNING
-80 ORIENTATION, FAMILIARIZATION
-81 TASK NOMENCLATURE, IDENTS, LOCATIONS, FACTS, RULES
83 PROCEDURES AT VERBAL LEVEL ONLY
84 TASK COMPONENTS WITH GUIDANCE
85 ENTIRE JOB-TASK PROCEDURALLY AT BARELY ACCEPTABLE MASTERY
86 HIGHLY PROFICIENT IN WORK CONTEXT
87 UNUSUAL TASK CONDITIONS
```

```
** MAKE NEXT ACTION REQUEST
_DELETE 87
** COURSES  DESIGNATOR NUMBER       87 HAS BEEN DELETED

** MAKE NEXT ACTION REQUEST
_ADD 82

** ADDED DESIGNATOR = 82 TASK FORMATS AT CONCEPTUAL LEVEL
** TOTAL DESIGNATORS NOW SELECTED :         11

** MAKE NEXT ACTION REQUEST
_FILE

** MODIFIED COURSES  DESIGNATORS HAVE BEEN SAVED

SEARCH DESIGNATOR EDITOR NOW TERMINATING
```

PROGRAM NAME : P3

ENTRY POINT : P3

RUN EXECUTIVES(S) : RUNP3, NONAME, IFNAME, JCL, JCL2

USER OUTPUT VIA : Terminal Only

USER PROMPTING : Not Required

FUNCTION : Range-Of-Effects Search Module

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
MREF	INPUT	1	DASD	ETAM/MREF	F, 80, 80
DIND	DIRECT	3	DASD	ETAM/DIND	F, 80, 80
SARG	INPUT	30	DASD	Projectname/SARG	F, 240, 240
RESU	OUTPUT	31	DASD	Projectname/RESU	FB, 16, 800
ID	INPUT	26	DASD	Projectname/ID	F, 80, 80
COURSE	DIRECT	7	DASD	ETAM/COURSE	F, 80, 80
DCIN	INPUT	8	DASD	ETAM/DCIN	F, 760, 760
VEHS	DIRECT	15	DASD	ETAM/VEHS	F, 80, 80
DVEH	INPUT	16	DASD	ETAM/DVEH	F, 750, 750
TASKS	DIRECT	22	DASD	ETAM/TASKS	F, 80, 80
DTASK	INPUT	23	DASD	ETAM/DTASK	F, 1500, 1500
DESC	INPUT	6	DASD	ETAM/DESC	F, 213, 213
VDESC	INPUT	12	DASD	ETAM/VDESC	F, 213, 213
TDESC	INPUT	19	DASD	ETAM/TDESC	F, 213, 213

Discussion

The Range-of-Effects search process is accomplished within Program P3. The descriptor search arguments are input via File Number 30. As mentioned in the section covering Program P2, File Number 30 is created only through a successful execution of Program P2.

Only synopsis type output is directed to the online terminal as a result of executing Program P3. The search results output is passed via filetype Program P3. The search results output is passed via filetype RESU (File Number 31) to the Programs P5A, P5B, and P5C for output editing and processing.

Listing of EXECUTIVE = RUNP3

```
      &TYPE OFF
      &COMMENT RANGE-OF-EFFECT (ROE) SEARCH EXECUTION.
      &COMMENT SINGLE ENTRY PARM REQUIRED IS PROJECT NAME.
      &COMMENT OUTPUT ONLY TO TERMINAL IN THIS SEQUENCE.
      &IF &INDEX EQ 1 &GOTO -STP1
  EXEC NONAME RUNP3
  -STP1 &ALPHA3 = &1
  EXEC IFNAME
  EXEC JCL
  EXEC JCL2
  LOAD P3 (CLEAR LIBE) PLILIB
  START (BRIEF)
  FILEDEF * CLEAR
      &EXIT
```

Program P3 Specific User Instructions

The Project Files to be referenced by Program P3 in a single unique run can only be the files associated with a single ETAM project. The name of the desired project must be supplied by the User when invoking the RUNP3 executive. On the sample terminal outputs shown below, the project being referenced is "PN123".

Only the project name must be supplied by the User when initiating a Program P3 run. As mentioned above, Program P3 cannot be run until Program P2 has been successfully run at least one time.

Program P3 Error Messages

The following general error text can be output from Program P3:

```
** ERROR NUMBER  XX  HAS OCCURED **
```

(followed by a general data-output line)

where the error number "XX" in the above message has the following significance. All are internal program indexing errors.

01 - Indexing error in handling categories within the Master Descriptor Index file (File Number 3).

02 - Total number of input search descriptors exceeds 99. Note that no descriptor is greater than a two-digit number.

03 - More than 15 categories of descriptors found for a single entity within the Master Descriptor Index.

04 - Attempted to access a record in the Abbreviated Course file (File Number 7) beyond the bounds of the file.

05 - Same error type as type 04, except for the Abbreviated Vehicle file (File Number 15).

06 - Same error type as type 04, except for the Abbreviated Tasks file (File Number 22).

Program P3 Sample Run Output

The following reflects a complete run of Program P3 against the sample project "PN123".

```
              RUNP3 PN123

   PROJECT NAMED PN123 ALREADY EXISTS,
   IS THIS CORRECT? (RESPOND YES OR QUIT)

   YES
   $$$
   RANGE-OF-EFFECTS (ROE) SEARCH PROGRAM IS STARTING

   RANGE-OF-EFFECTS SEARCH FOR PROJECT : PN123
   3-D PROCEDURAL TRAINER

   *** NOW PROCESSING COURSES  SEARCH -  11 DESCRIPTORS SELECTED

           TOTAL NUMBER DESCRIPTOR RECORDS EXAMINED :     122
         TOTAL RECORDS SELECTED WITH MATCHING DATA :      37
   TOTAL RECORDS SELECTED WITHOUT ABBREVIATED DATA :      24
       ACCUMULATED SEARCH OUTPUT RECORDS, THUS FAR :      61
```

```
*** NOW PROCESSING VEHICLES SEARCH -    0 DESCRIPTORS SELECTED

*** NO SEARCH DESCRIPTORS SPECIFIED FOR VEHICLES TYPE DATA - SEARCH IS BYPASSED

*** NOW PROCESSING TASKS    SEARCH -    0 DESCRIPTORS SELECTED

*** NO SEARCH DESCRIPTORS SPECIFIED FOR TASKS    TYPE DATA - SEARCH IS BYPASSED

*** DESIGNATOR SEARCH PROGRAM IS TERMINATING

    TOTAL ACCUMULATED SEARCH OUTPUT RECORDS :     61
```

PROGRAM NAME : P5A

ENTRY POINT : IHECMS (Execution Parameter Supplied by Executive
Up on Entry)

RUN EXECUTIVES(S) : RUNP5A, NONAME, IFNAME, DEVICE, JCL, JCL2

USER OUTPUT VIA : Terminal, Offline Printer, or Both

USER PROMPTING : External Executive - Output Device Select

FUNCTION : Print Results of Range-Of-Effects Search

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
ID	INPUT	26	DASD	Projectname/ID	F, 80, 80
RESU	INPUT	31	DASD	Projectname/RESU	FB, 16, 800
MREF	INPUT	1	DASD	ETAM/MREF	F, 80, 80
COURSE	DIRECT	7	DASD	ETAM/COURSE	F, 80, 80
DCIN	INPUT	8	DASD	ETAM/DCIN	F, 760, 760
DCDP	INPUT	9	DASD	ETAM/DCDP	F, 1200, 1200
VEHS	DIRECT	15	DASD	ETAM/VEHS	F, 80, 80
DVEH	INPUT	16	DASD	ETAM/DVEH	F, 750, 750
TASKS	DIRECT	22	DASD	ETAM/TASKS	F, 80, 80
DTASK	INPUT	23	DASD	ETAM/DTASK	F, 1500, 1500
SYSPRINT	OUTPUT	N/A : THIS IS THE OFFLINE PRINTER			FA, 121, 121

Discussion

The direct results of the Range-of-Effects search (Program P3) are printed by this program.

Listing of EXECUTIVE = RUNP5A

```
          &TYPE OFF
          &COMMENT ROE PAW SEARCH RESULTS PPINTED HEREIN.
          &COMMENT SINGLE ENTRY PARM REOUIRED - PROJECT NAME
          &COMMENT EXEC = DEVICE WILL PROMPT FOR OUTPUT DEVICE
          &IF &INDEX EO 1 &GOTO -STP1
    EXEC NONAME RUNP5A
    -STP1 &ALPHA3 = &1
    EXEC IFNAME
          &SPACE
    EXEC DEVICE
          &SPACE
    EXEC JCL
    EXEC JCL2
    FILEDEF SYSPRINT PTR RE FA LR 121 BL 121
    LOAD P5A (CLEAR LIBE) PLILIB
    START IHECMS &ALPHA1 (BRIEF)
    FILEDEF * CLEAR
          &EXIT
```

Program P5A Specific User Instructions

A single run of Program P5A processes only those Project Files associated with a single unique project. The name of the project of concern must be provided by the User when invoking the RUNP5A executive. In the sample output provided below, this is illustrated by reference to project "PN123".

The Range-of-Effects search program (Program P3) provides no direct output, only summary information. This program (Program P5A) produces an itemized listing of the search results using the RESU (File Number 31) as input.

The output records are numbered. These numbers will provide reference for any editing activity that might follow by running Program P5B.

Program P5A Error Messages

None generated.

Program P5A Sample Run OUtput

The following reflects a complete run of Program P5A, using the sample project name "PN123".

```
            RUNP5A PN123

    PROJECT NAMED PN123 ALREADY EXISTS,
    IS THIS CORRECT? (RESPOND YES OR QUIT)

    YES


    OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
    ENTER TERM, PRTR, BOTH, OR QUIT
    BOTH

    $$EXECUTION:

             :::: INITIAL SEARCH RESULTS FOR PROJECT -PN123       ::::
               3-D PROCEDURAL TRAINER





         PAGE NUMBER :      1

    RANGE-OF-EFFECTS RESULTS FOR : COURSES
    ----------------------------------------

         1 A1020060  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
         2 A1020093  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
         3 A1020095  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
         4 A1930050 2213 INERT NAV PRINC
         5 A2420010 6529 IS A
         6 A4120010 6078 EA-A
         7 A4120010 6287 EA A
         8 A4310011 1035 EOD BASIC NAVY
         9 A4310014 1036 EOD REFR NAVY
        10 A4330019 2040 MASTER DIVQUAL
        11 A4330025 2032 DIVER FIRST
        12 A4910014 204G RADIUM REMOV OPS
        13 A5510019 0133 CAR HAND BAS
        14 A5510027 3931 UNREP MECH/HYD
        15 A5510068 0134 FORK LIFT OPER
        16 A5800016 6053 CTO A
        17 A6510020 5399 1200 PSI BT
        18 A6520050 5224 OX GENR 6L160P
        19 A6520068 463E SOLAR GAS TURBIN
```

```
20 A6700011 3078 WATCH REPAIR
21 A7010027 337M WELD/HPRES PIPE
22 A7200013 466Z UT-J
23 B3030051  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
24 C 003722 2887 TA4JF FAM PILOTS
25 C 003722 9007 TA4JF FAM PILOTS
26 C 003722 9009 TA4JF FAM PILOTS

27 C 003722 9010 TA4JF FAM PILOTS
28 C 003722 9738 TA4JF FAM PILOTS
29 C 2C3352 2576 UH1N A/C FAM/P/
30 C1003834 7529 AAS18 INT MAI
31 C1023793 7914 A7AB ATT HEADING
32 C1023793 7915 A7AB ATT HEADING
33 C1213011 346L AWM23 RADIO FREQ
34 C1213011 544L AWM23 RADIO FREQ
35 C6003472 340C E2C WEA SYS FAM
36 C6023536 341D 54H6077 PROP INT
37 C6463103 7365 CVA/CV ALW SUPV
38 D 2C0011  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
39 E 2A1001  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
40 E 2A1301  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
41 E 2A1803  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
42 E 2C0901  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
43 E 2D0016  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
44 E 2D0075  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
45 E6010210  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
46 E6461641  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
47 F7000010  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
48 H 2E3710  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
49 J1300645  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
50 J2210357 538Q MULTI-THREAT TNG
51 J2330203 2637 ADV EW OP'S CRSE
52 J2430974 2181 IPC
53 J8000433  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
54 K2330066 205Y SUB EW OP-ADV
55 L1010024  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN




  PAGE NUMBER :     2

RANGE-OF-EFFECTS RESULTS FOR : COURSES
-------------------------------------

    56 L6610056  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
    57 M198100E  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
    58 N7010320  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
    59 Q 2C0015  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
    60 83000010  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
    61 83000012  NOTE - NO MATCHING NITRAS DATA AVAILABLE FOR THIS CIN
```

```
*** RESULTS FILE NOW AT END-OF-FILE ***

   TOTAL NUMBER COURSE RECORDS PROCESSED :       61
  TOTAL NUMBER VEHICLE RECORDS PROCESSED :        0
  TOTAL NUMBER JOBTASK RECORDS PROCESSED :        0
     TOTAL NUMBER ROE RESULT RECORDS READ :       61
RECORDS WITHOUT MATCHING ABBREVIATED DATA :       24
  TOTAL NUMBER DIRECTORY SEARCH FAILURES :        0
```

PROGRAM NAME : P5B
ENTRY POINT :
RUN EXECUTIVES(S) : RUNP5B, NONAME, IFNAME, JCL, JCL2
USER OUTPUT VIA : Terminal Only
USER PROMPTING : Internal Program Only
FUNCTION : Interactive Edit of Range-Of-Effects (ROE) Results

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
ID	INPUT	26	DASD	Projectname/ID	F, 80, 80
REE	OUTPUT	32	DASD	Projectname/REE	FB, 16, 800
RESU	INPUT/ OUTPUT	31	DASD	Projectname/RESU	FB, 16, 800
MREF	INPUT	1	DASD	ETAM/MREF	F, 80, 80
COURSE	DIRECT	7	DASD	ETAM/COURSE	F, 80, 80
DCIN	INPUT	8	DASD	ETAM/DCIN	F, 760, 760
DCDP	INPUT	9	DASD	ETAM/DCDP	F, 1200, 1200
TASKS	DIRECT	22	DASD	ETAM/TASKS	F, 80, 80
DTASK	INPUT	23	DASD	ETAM/DTASK	F, 1500, 1500
VEHS	DIRECT	15	DASD	ETAM/VEHS	F, 80, 80
DVEH	INPUT	16	DASD	ETAM/DVEH	F, 750, 750

Discussion

This program allows the User to peruse and edit the results of a previous Range-of-Effects search. In this regard:

- Program P3 is the search routine. This module creates the un-edited results file RESU (File Number 31).
- Program P5A prints the RESU file and numbers the records therein. These record numbers are required for reference in the operation of Program P5B.

Program P5B modifies the search results RESU as desired by the User. A modified file, containing only original search results and added records is created by this program; this is the REE file (File Number 32).

Listing of EXECUTIVE = RUNP5B

```
     &TYPE OFF
     &COMMENT INTERACTIVE EDIT OF THE POE SEARCH RESULTS.
     &COMMENT SINGLE ENTRY PARM REQUIRED IS PROJECT NAME.
     &COMMENT OUTPUT ONLY TO TERMINAL IN THIS SEQUENCE.
     &IF &INDEX EQ 1 &GOTO -STP1
 EXEC NONAME RUNP5B
 -STP1 &ALPHA3 = &1
 EXEC IFNAME
 EXEC JCL
 EXEC JCL2
 LOAD P5B (CLEAR LIBE) PLILIB
 START (BRIEF)
 FILEDEF * CLEAR
     &EXIT
```

Program P5B Specific User Instructions

Program P5B processes only those files associated with a unique ETAM project during a single run. It is mandatory that the name of the Specific project be provided by the User when invoking the RUNP5B executive. On the sample runs to follow, this is illustrated using the ETAM project name "PN123".

After initialization, Program P5B returns to the following general prompt message:

```
** MAKE NEXT ACTION REQUEST
```

This invitation for entry can be followed by one of the following entries:

```
DELETE  n1

DELETE  n1   n2

PRINT   n1

PRINT   n1   n2

ADDC    /cin/cdp/

ADDV    /stocknumber/

ADDT    /rate/rank/jobtask/

QUIT
```

The "DELETE" entries allow deletion of one or a range of entities in the search results file. The numbers "n1", "n2" refer to the numbered records in the printer listing of the search results as provided by Program P5A.

The "PRINT" commands provide a printed output of the record(s) indicated.

To add an entity not currently in the search results, the "ADDC", "ADDV", and "ADDT" entries are used for adding a Course, Vehicle, or Task, respectively. The Vehicle "stocknumber" must be present for an "ADDV" entry. For an "ADDC" entry, only the "cin" portion is required. When only "cin" is provided, all Courses bearing that "cin" designation will be selected from the Abbreviated Course data base. For the "ADDT" entry, only a single field is required. To match entries in the Abbreviated Tasks data base, all three must be provided.

The "QUIT" option rewrites the RESU file (File Number 31) with all entries - original, deleted, new (added). When printing the RESU file using Program P1, the status of each entry will be indicated. The Extract Results file (File Number 32) is composed by Program P5B. Only original and added entities are copied into this file.

Program P5B Error Messages

The following error text results from an illegal response to the general prompt message:

```
** CANNOT DECODE LAST ENTRY  -  TRY AGAIN
```

(followed by the general prompt message)

Sufficient internal table space has been reserved within Program P5B to allow a total of 500 new entity additions. Should an attempt be made to add more than this total, the following results:

```
** A TOTAL OF 500 NEW RECORDS HAVE BEEN ADDED
   ANOTHER RUN OF THIS ROUTINE MUST BE MADE TO
         ENTER FURTHER ADDITIONS.
```

At this point, if the "QUIT" option is elected, the Program will rewrite the results files and terminate. A second invocation of Program P5B will then allow 500 more additions to be made.

When entering record numbers in conjunction with the "DELETE" and "PRINT" options, two errors can occur. One is the addressing of a record that is beyond the limits of the results file RESU. A second error can occur when using a range-type entry, e.g., DELETE n1 n2. If n1 is greater in value than n2, or either is outside the bounds of the RESU file, the following common message results:

```
** RANGE ERROR IN RECORD NUMBER
```

Program P5B Sample Run Output

The following terminal output sample illustrates the common entries in the use of Program P5B.

```
          RUNP5B PN123

 PROJECT NAMED PN123 ALREADY EXISTS.
 IS THIS CORRECT? (RESPOND YES OR QUIT)

 YES
 $$$$
          ::: ROE SEARCH RESULTS EDIT FOR PROJECT PN123      :::
              3-D PROCEDURAL TRAINER

 :::    61 HAVE BEEN LOADED INTO MEMORY FOR EDITING

 ::: MAKE NEXT ACTION REQUEST
 _DELETE 1 3
 DELETED RECORD NUMBER : 1     THRU 3

 ::: MAKE NEXT ACTION REQUEST
 _DELETE 23
 DELETED RECORD NUMBER : 23

 ::: MAKE NEXT ACTION REQUEST
 _DELETE 38 49
 DELETED RECORD NUMBER : 38    THRU 49

 ::: MAKE NEXT ACTION REQUEST
 _DELETE 53
 DELETED RECORD NUMBER : 53

 ::: MAKE NEXT ACTION REQUEST
 _DELETE 55
 DELETED RECORD NUMBER : 55

 ::: MAKE NEXT ACTION REQUEST
 _ADDC /A1010108/

 THE FOLLOWING RECORD(S) HAS/HAVE BEEN ADDED...

          RECNO/CIN/CDP/COURSE :     62/A10101087654/ WRA-4 CMB MA
```

```
** MAKE NEXT ACTION REQUEST
_PRINT 1 61

RECNO/CIN/CDP/TITLE :       1/A1020060/     / (NO MATCHING DATA)

RECNO/CIN/CDP/TITLE :       2/A1020093/     / (NO MATCHING DATA)

RECNO/CIN/CDP/TITLE :       3/A1020095/     / (NO MATCHING DATA)

RECNO/CIN/CDP/TITLE :       4/A1930050/2213/ INERT NAV PRINC

RECNO/CIN/CDP/TITLE :       5/A2420010/6529/ IS A

RECNO/CIN/CDP/TITLE :       6/A4120010/6078/ EA-A

RECNO/CIN/CDP/TITLE :       7/A4120010/6287/ EA A
```

... Output interrupted at this point ...

```
RECNO/CIN/CDP/TITLE :       58/N7010320/    / (NO MATCHING DATA)

RECNO/CIN/CDP/TITLE :       59/Q 2C0015/    / (NO MATCHING DATA)

RECNO/CIN/CDP/TITLE :       60/83000010/    / (NO MATCHING DATA)

RECNO/CIN/CDP/TITLE :       61/83000012/    / (NO MATCHING DATA)

** MAKE NEXT ACTION REQUEST
_QUIT

** ROE SEARCH RESULTS EDITOR IS TERMINATING

   TOTAL RECORDS ADDED :    1

   TOTAL RECORDS IN PROJECT EXTRACT FILE :     44
```

PROGRAM NAME : P5C

ENTRY POINT : IHECMS (Runtime Parameter Supplied by Executive Upon Entry)

RUN EXECUTIVES(S) : RUNP5C, NONAME, IFNAME, DEVICE, JCL, JCL2, SORT5C

USER OUTPUT VIA : Terminal, Offline Printer, or Both

USER PROMPTING : At Executive Level : Output Device Select

FUNCTION : Print Edited Results of Range-of-Effects (ROE) Search

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
SYSPRINT	OUTPUT	N/A : THIS IS THE OFFLINE PRINTER			F, 121, 121
REE	INPUT	32	DASD	Projectname/REE	FB, 16, 800
ID	INPUT	26	DASD	Projectname/ID	F, 80, 80
MREF	INPUT	1	DASD	ETAM/MREF	F, 80, 80
COURSE	DIRECT	7	DASD	ETAM/COURSE	F, 80, 80
DCIN	INPUT	8	DASD	ETAM/DCIN	F, 760, 760
DCDP	INPUT	9	DASD	ETAM/DCDP	F, 1200, 1200
VEHS	DIRECT	15	DASD	ETAM/VEHS	F, 80, 80
DVEH	INPUT	16	DASD	ETAM/DVEH	F, 750, 750
TASKS	DIRECT	22	DASD	ETAM/TASKS	F, 80, 80
DTASK	INPUT	23	DASD	ETAM/DTASK	F, 1500, 1500

Discussion

Program P5C is the last step in the Range-of-Effects (ROE) search process. The edited results of a search are sorted and printed. The record sort is executed at the executive level in executive SORT5C.

Listing of EXECUTIVE = RUNP5C

```
       &TYPE OFF
       &COMMENT ROE EXTRACT SEARCH RESULTS PRINTED HEREIN.
       &COMMENT SINGLE ENTRY PARM REOUIRED - PROJECT NAME
       &COMMENT EXEC = DEVICE WILL PROMPT FOR OUTPUT DEVICE
       &IF &INDEX EO 1 &GOTO -STP1
  EXEC NONAME RUNP5C -
  -STP1 &ALPHA3 = &1
  EXEC IFNAME
       &SPACE
  EXEC DEVICE
       &SPACE
  EXEC JCL
  EXEC JCL2
  EXEC SORT5C
  FILEDEF SYSPRINT PTR RE FA LP 121 BL 121
  LOADMOD P5C
  START IHECMS &ALPHA1 (BRIEF)
  FILEDEF * CLEAR
       &EXIT
```

Listing of EXECUTIVE = SORT5C

```
     &TYPE OFF
     &COMMENT SORT SUPPORT FOR ROE RESULTS FN/REE PROJECT FILE.
     &COMMENT NO EXPLICIT ENTRY PARMS PEOUIRED FOR THIS EXEC.
     &COMMENT ALPHA3 GLOBAL CONTAINS PROJECT FILENAME ON ENTRY.
     &DSN = . !! &ALPHA3
STATE &DSN REE P
     &IF &INDEX0 NE 0 &GOTO -STPS1
ERASE &DSN REE P
-STPS1 STATE &ALPHA3 REE P
     &IF &INDEX0 EO 0 &GOTO -STPS2
     &SPACE
     &PRINT PROJECT FILE &ALPHA3 / REE DOES NOT EXIST!
-SER2 &SPACE
     &PRINT EYEC SEOUENCE RUNP5C IS TERMINATING ABNORMALLY...
     &SPACE
     &OUIT
-STPS2 &STACK 1 1 4 16
     &STACK KT
SORT &ALPHA3 PEE (BPIEF)
     &SRET = &INDEX0
     &PRINT &SRET &INDEX0 &ALPHA3
    &SPACE
     &IF &SRET EO 0 &GOTO -STPS3
```

```
-SER1 &PRINT SORT RETURN CODE WAS &SRET
      &GOTO -SER2
-STPS3 STATE &DSN REE P
      &IF &INDEX0 EQ 0 &GOTO -STPS4
      &PRINT SORT OUTPUT FILE DOES NOT EXIST
      &GOTO -SER1
-STPS4 ERASE &ALPHA3 REE P
ALTER &DSN REE P &ALPHA3 REE P
ERASE &DSN REE P
      &EXIT
```

Program P5C Specific User Instructions

Each run of Program P5C is associated with a unique ETAM project results file. Accordingly, the name of the project of concern must be provided by the User when invoking the RUNP5C executive. On the sample run to follow, this has been accomplished using the project name "PN123".

Program P5C Error Messages

The following error message text can result from running Program P5C:

```
** ERROR NUMBER  XX HAS OCCURED **
```

(followed by a general error data output line)

This message is significant for only the following error number "XX":

01 - The REE input file (File Number 32) cannot contain deleted entities. This error occurs if such an entity is passed into the REE file by Program P5B.

Program P5C Sample Run Output

The following is a complete run of Program P5C for project "PN123".

```
        RUNP5C PN123

PROJECT NAMED PN123 ALREADY EXISTS,
IS THIS CORRECT? (RESPOND YES OR QUIT)

YES
```

```
OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
ENTER TERM, PRTR, BOTH, OR QUIT
BOTH

$$0 0 PN123

EXECUTION:

         *** EXTRACT SEARCH RESULTS FOR PROJECT - PN123       ***
           3-D PROCEDURAL TRAINER




     PAGE NUMBER :      1

   1  ORIGINAL  A1930050  2213  INERT NAV PRINC
   2  ORIGINAL  A2420010  6529  IS A
   3  ORIGINAL  A4120010  6078  EA-A
   4  ORIGINAL  A4120010  6287  EA A
   5  ORIGINAL  A4310011  1035  EOD BASIC NAVY
   6  ORIGINAL  A4310014  1036  EOD REFR NAVY
   7  ORIGINAL  A4330019  2040  MASTER DIVQUAL
   8  ORIGINAL  A4330025  2082  DIVER FIRST
   9  ORIGINAL  A4910014  204G  RADIUM REMOV OPS
  10  ORIGINAL  A5510019  0133  CAR HAND BAS
  11  ORIGINAL  A5510027  3931  UNREP MECH/HYD
  12  ORIGINAL  A5510068  0134  FORK LIFT OPER
  13  ORIGINAL  A5800[illegible]6  6053  CTO A
  14  ORIGINAL  A6510020  5399  1200 PSI BT
  15  ORIGINAL  A6520050  5224  OX GENR 6L160P
  16  ORIGINAL  A6520068  463E  SOLAR GAS TURBIN
  17  ORIGINAL  A6700011  3078  WATCH REPAIR
  18  ORIGINAL  A7010027  337M  WELD/HPRES PIPE
  19  ORIGINAL  A7200013  466Z  UT-J
  20  ORIGINAL  C 003722  2887  TA4JF FAM PILOTS
  21  ORIGINAL  C 003722  9007  TA4JF FAM PILOTS
  22  ORIGINAL  C 003722  9009  TA4JF FAM PILOTS
  23  ORIGINAL  C 003722  9010  TA4JF FAM PILOTS
  24  ORIGINAL  C 003722  9738  TA4JF FAM PILOTS
  25  ORIGINAL  C 2C3352  2576  UH1N A/C FAM/P/
  26  ORIGINAL  C1003834  7529  AAS18 INT MAI
  27  ORIGINAL  C1023793  7914  A7AB ATT HEADING
  28  ORIGINAL  C1023793  7915  A7AB ATT HEADING
  29  ORIGINAL  C1213011  346L  AWM23 RADIO FREQ
  30  ORIGINAL  C1213011  544L  AWM23 RADIO FREQ
  31  ORIGINAL  C6003472  340C  E2C WEA SYS FAM
  32  ORIGINAL  C6023536  341D  54H6077 PROP INT
```

```
 33  ORIGINAL  C6463103  7365  CVA/CV.ALW SUPV
 34  ORIGINAL  J2210357  538Q  MULTI-THREAT TNG
 35  ORIGINAL  J2330203  2637  ADV EW OP'S CRSE
 36  ORIGINAL  J2430974  2181  IPC
 37  ORIGINAL  K2330066  205Y  SUB EW OP-ADV
 38  ORIGINAL  L6610056  NONE  (NO ABBREVIATED FILE DATA AVAILABLE)
 39  ORIGINAL  M198100E  NONE  (NO ABBREVIATED FILE DATA AVAILABLE)
 40  ORIGINAL  N7010320  NONE  (NO ABBREVIATED FILE DATA AVAILABLE)
 41  ORIGINAL  Q 2C0015  NONE  (NO ABBREVIATED FILE DATA AVAILABLE)
 42  ORIGINAL  83000010  NONE  (NO ABBREVIATED FILE DATA AVAILABLE)
 43  ORIGINAL  83000012  NONE  (NO ABBREVIATED FILE DATA AVAILABLE)
 44   ADDED    A1010108  7654  (NO ABBREVIATED FILE DATA AVAILABLE)

*** NOW AT EOF ON SEARCH RESULTS FILE ***

     TOTAL NUMBER COURSE RECORDS PROCESSED :      44
    TOTAL NUMBER VEHICLE RECORDS PROCESSED :       0
    TOTAL NUMBER JOBTASK RECORDS PROCESSED :       0
      TOTAL NUMBER ROE RESULT RECORDS READ :      44
RECORDS WITHOUT MATCHING ABBREVIATED DATA :       7
    TOTAL NUMBER DIRECTORY SEARCH FAILURES :       0
```

PROGRAM NAME : P7

ENTRY POINT : P7

RUN EXECUTIVES(S) : RUNP7

USER OUTPUT VIA : Terminal Only

USER PROMPTING : None

FUNCTION : Makeup Course CIN Directory (File Number 8) and CDP Directory (File Number 9)

NOTE : This program contains an internal SORT invocation

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
MREF	INOUT	1	DASD	ETAM/MREF	F, 80, 80
COURSE	INPUT	7	DASD	ETAM/COURSE	F, 80, 80
DCIN	OUTPUT	8	DASD	ETAM/DCIN	F, 760, 760
DCDP	OUTPUT	9	DASD	ETAM/DCDP	F, 1200, 1200
SORTWK01 SORTWK02 SORTWK03	SORT	N/A	DASD (TEMP)	Standard internal SORT work areas	

Discussion

This module constructs the two directories associated with the Abbreviated Course file. The Course file itself is used as the primary input. The directories will consist of Course CDP numbers and record pointers in one file and Course CIN numbers and pointers in the other.

Each directory is sorted by its key field (CIN or CDP); a PL/1 invocation of the standard PL/1 sort package is included within Program P7. Some of the messages output from this module are the standard Sort/Merge messages and warnings.

Listing of EXECUTIVE = RUNP7

```
        &TYPE OFF
        &COMMENT MAKEUP OF COURSE DIRECTORIES (CIN AND CDP)
        &COMMENT NO ENTRY PARMS
   FILEDEF COURSE DSK ETAM COURSE RE F LR 80 BL 80
   FILEDEF MREF DSK ETAM MREF RE F LR 80 BL 80
   FILEDEF DCIN DSK ETAM DCIN RE F LR 760 BL 760
   FILEDEF DCDP DSK ETAM DCDP RE F LR 1200 BL 1200
   ATTACH TEMP5 AS 192 BRIEF
   FILEDEF SORTWK01 DSK W1 SORT T
   FILEDEF SORTWK02 DSK W2 SORT T
   FILEDEF SORTWK03 DSK W3 SORT T
   LOAD P7 (CLEAR LIBE) PLILIB
   START (BRIEF)
   FILEDEF * CLEAR
   DETACH 192 BRIEF
        &EXIT
```

Program P7 Error Messages

In the event of an error detected by the Sort/Merge package:

```
** SORT RETURN CODE INVALID, CODE WAS 16 **

COURSE DIRECTORY CONSTRUCT PROGRAM IS NOW TERMINATING
```

The following message strings are caused by overflow of the internal table allotted for directories. The current implementation allots a CIN directory maximum of 304 entries, and a CDP directory max of 1200 entries. One of the message sequences below will result in the event of overflow:

```
**** CIN TABLE OVERFLOW ****
```

/or/

```
**** CDP TABLE OVERFLOW ****

DIRECTORY CONSTRUCT PROGRAM IS TERMINATING DUE TO ERROR
** SORT RETURN CODE INVALID, CODE WAS 8 **

COURSE DIRECTORY CONSTRUCT PROGRAM IS NOW TERMINATING
```

Program P7 Specific User Instructions

None.

Program P7 Sample Run Output

The following represents an actual run of P7. The output is shown in its entirety. Note that this output is directed only to the online terminal printer device. The output below represents the actual contents of both the Course CIN and CDP directories in the current ETAM implementation.

```
          RUNP7
$
ABBREVIATED COURSE DATABASE CIN/CDP DIRECTORY
     CONSTRUCT PROGRAM

IER036I - B =  363
IER037I - G =  543
IER038I - NMAX =    11970
::: EOF ON COURSE INPUT FILE ; SORT PHASE START

IER045I - END SORT PH
IER049I - SKIP MERGE PH
::: SORT PHASE COMPLETE ; STARTING CIN DIRECTORY MAKEUP

CIN NUM    CIN       TOTAL CDPS   REC PTR
-------  --------   ----------   -------

      1  A 000111           1         0
      2  A 2E0013           1         1
      3  A 2G0014           1         2
      4  A 8C0015           1         3
      5  A1010108           3         6
      6  A1930050           1         7
      7  A2420010           1         8
      8  A4120010           2         9
      9  A4310011           1        11
     10  A4310014           1        12
     11  A4330019           1        13
     12  A4330025           1        14
     13  A4910014           1        15
     14  A5000011           2        17
     15  A5000028           2        19
     16  A5000032           4        23
     17  A5100012           1        24
     18  A5320015           1        25
     19  A5420014           1        26
     20  A5510019           1        27
     21  A5510027           1        28
     22  A5510068           1        29
     23  A5700010           1        30
     24  A5800016           1        31
     25  A6510010           2        33
     26  A6510020           2        35
     27  A6520050           1        36
     28  A6520068           1        37
     29  A6700011           1        38
     30  A6700025           1        39
     31  A7010027           2        41
     32  A7020024           1        42
```

```
 33     A7100016        1       43
 34     A7120011        2       45
 35     A7200013        2       47
 36     C 003491        1       48
 37     C 003722        5       49
 38     C 003807        1       54
 39     C 2A3502        1       55
 40     C 2A3531        1       56
 41     C 2A3831        4       57
 42     C 2C3352        1       61
 43     C1003834        1       62
 44     C1023793        2       63
 45     C1213011        2       65
 46     C2222010        1       67
 47     C4202011        1       68
 48     C5163203        6       69
 49     C6003358        1       75
 50     C6003472        1       76
 51     C6003831        2       77
 52     C6023353        1       79
 53     C6023482        2       81
 54     C6023536        2       83
 55     C6463103        3       86
 56     J 2E0100        1       87
 57     J2210303        2       89
 58     J2210310        1       90
 59     J2210344        2       91
 60     J2210348        1       93
 61     J2210357        2       95
 62     J2330203        1       96
 63     J2330211        5       97
 64     J2430974        1      102
 65     J2500313        2      103
 66     J6440914        1      105
 67     J6520474        1      106
 68     K 2E1078        1      107
 69     K2210042        1      108
 70     K2220035        1      109
 71     K2330066        1      110

*** LAST OF CIN DIRECTORY ENTRIES ENCOUNTERED ***

*** NOW STARTING CDP DIRECTORY ***

CDP NUM  CDP   REC PTR
-------  ----  -------

   1     0133     27
   2     0134     29
   3     0143      2
   4     0303      1
   5     0327      0
   6     1034    105
   7     1035     11
```

8	1036	12
9	201O	20
10	202M	21
11	204E	80
12	204G	15
13	2040	13
14	205W	97
15	205Y	110
16	2082	14
17	2121	39
18	2131	3
19	2181	102
20	2213	7
21	2374	57
22	2398	18
23	2576	61
24	2601	22
25	2637	96
26	265V	103
27	266C	104
28	2667	81
29	2671	82
30	2717	69
31	2887	49
32	3078	38
33	3192	16
34	3193	17
35	3202	40
36	3205	43
37	321X	70
38	321Z	71
39	322B	72
40	322D	73
41	322E	74
42	3249	30
43	337M	41
44	340C	76
45	341D	83
46	346F	44
47	346L	65
48	349H	84
49	349K	85
50	351J	42
51	351Q	45
52	3931	28
53	4028	46
54	4138	34
55	4510	68
56	461W	19
57	463E	37
58	466Z	47
59	470P	25
60	4717	26

```
     61      4964       75
     62      5160        4
     63      5224       36
     64      5250       98
     65      526K       91
     66      528E       88
     67      528H       92
     68      528J       93
     69      5324       99
     70      533P       94
     71      533Z      100
     72      534A       89
     73      534Z      109
     74      538P       90
     75      538Q       95
     76      539J       23
     77      5399       35
     78      540M       58
     79      540N       59
     80      542N      101
     81      544L       66
     82      5535        5
     83      5599      108
     84      6053       31
     85      6057       24
     86      6069       32
     87      6078        9
     88      6260       33
     89      6278       67
     90      6287       10
     91      6529        8
     92      7365       86
     93      7529       62
     94      7540       77
     95      7541       78
     96      7654        6
     97      7914       63
     98      7915       64
     99      7933       79
    100      8178      106
    101      9007       50
    102      9009       51
    103      9010       52
    104      9033       54
    105      9051       55
    106      9062       60
    107      9279       87
    108      9738       53
    109      9740      107
    110      9750       48
    111      9751       56
IER055I - INSERT        222,DELETE       222
IER054I - RCD IN           ,OUT
IER052I - EOJ
::: SORT COMPLETED OK, CODE = 0 :::

::: CIN DIRECTORY HAS BEEN SUCCESSFULLY REWRITTEN

::: CDP DIRECTORY HAS BEEN SUCCESSFULLY REWRITTEN

COURSE DIRECTORY CONSTRUCT PROGRAM IS NOW TERMINATING
```

PROGRAM NAME : P7A

ENTRY POINT : P7A

RUN EXECUTIVES(S) : RUNP7A

USER OUTPUT VIA : Terminal Only

USER PROMPTING : None

FUNCTION : Load Course Descriptor File (File Number 6)

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
MREF	INOUT	1	DASD	ETAM/MREF	F, 80, 80
DIND	DIRECT	3	DASD	ETAM/DIND	F, 80, 80
DESC	OUTPUT	6	DASD	ETAM/DESC	F, 213, 213
DCARD	INPUT	5	DASD	ETAM/SCDESC	F, 80, 80

Discussion

The output of this program is the Course Descriptor file (File Number 6). Note that the same descriptor card input file (File Number 5) was used in Program P14 to select the appropriate Course file CIN numbers. Accordingly, there will be no errors of mismatched CIN numbers in the run of this Program P7A.

Listing of EXECUTIVE = RUNP7A

```
    &TYPE OFF
    &COMMENT COURSE DESCRIPTORS FILE LOAD
    &COMMENT NO ENTRY PARMS
FILEDEF MREF DSK ETAM MREF RE F LR 80 BL 80
FILEDEF DIND DSK ETAM DIND RE F LR 80 BL 80
FILEDEF DCARD DSK ETAM SCDESC RE F LR 80 BL 80
FILEDEF DESC DSK ETAM DESC RE F LR 213 BL 213
LOAD P7A (CLEAR LIBE) PLILIB
START (BRIEF)
FILEDEF * CLEAR
    &EXIT
```

Program P7A Error Messages

The only error diagnosed by this program is that of an invalid numeric subscript occuring on an input card. This error is detected by comparing the given descriptor numbers with those indicated as legal in the Master Descriptor Index file (File Number 3). The following error message results in the event of a mis-compare; this error condition is not considered terminal, and the program continues with the next subscript for consideration.

```
** ERROR IN CIN -12345678- DESCRIPTOR SUBSCRIPT  999 DOES NOT EXIST **
```

Program P7A Specific User Instructions

None.

Program P7A Sample Run Output

The following represents an actual run of Program P7A. The output is not shown in its entirety. Note that this output is directed only to the online terminal printer device. The output shown represents the current contents of the Course Descriptor file.

```
          RUNP7A
5
COURSE FILE DESCRIPTOR LOAD PROGRAM

   CIN    ASSIGNED DESCRIPTOR(S)
-------- ----------------------

A 000111 01 03 04 11 12 13 20 21 22 33 42 45 48 53 65 66 82
A 2E0013 01 02 03 10 21 33 65 80 82
A 2G0014 01 02 03 12 33 41 42 44 45 48 52 53 65 69 82 84
A 4A0025 03 12 21 22 33 80
A 8C0015 01 05 12 20 32 80
A1010108 30 31 71 80 83
A1020060 03 31 64 65 71 85
A1020093 03
A1020095 05 11 12 13 20 21 31 64 65 82 83
A1930050 05 31 80
A2420010 05 20 21 22 30 32 63 64 65 66 85
A2440010 05 10 11 12 13 32 60 62 63 66 80
```

(Completion of text output presented on following page)

NOTE - Majority of text output has been eliminated for the sake of brevity. The last few lines of Program P7A output are as shown below.

```
K 002044 01 03 04 11 12 13 20 22 32 64 65 66 82 84
K 2E1078 01 02 03 05 10 12 21 33 42 45 49 53 65 66 72 82
K2210042 01 05 20 30 42 45 53 63 64 82 83
K2220035 02 03 20 30 41 44 49 52 61 62 64 67 69 72 86
K2330066 01 02 03 05 13 14 20 30 40 41 44 52 61 62 63 64 81 84
L1010024 05 80
L6610056 03 11 12 13 31 40 43 46 49 52 62 63 64 65 85
M198100E 03 11 20 31 40 44 46 49 52 64 65 84 85
N1000555 05 11 12 13 20 21 31 42 45 48 53 64 65 71 84
N7010320 03 05 10 11 12 13 20 31 40 44 46 49 51 61 62 63 64 85
P 003308 01 02 03 04 11 12 20 21 22 33 42 45 48 53 65 66 82 83
Q 2C0015 03 11 12 13 20 30 40 43 46 49 51 64 67 68 85
R2106104 05 20 32 42 45 48 64 83 89
S5000029 01 02 03 04 10 12 13 20 32 44 60 65 66 84
83000010 02 03 05 10 11 12 13 20 30 62 63 64 65 81 84
83000012 03 04 05 11 12 13 20 21 30 32 62 63 64 65 69 81 83

::: NOW AT EOF ON DESCRIPTOR CARD INPUT :::

    TOTAL NUMBER DESCRIPTOR INPUT CARDS :    132
      TOTAL UNIQUE CIN NUMBERS IN INPUT :    122
TOTAL NUMBER OUTPUT DESCRIPTOR RECORDS :    122
   TOTAL NUMBER CIN OR SUBSCRIPT ERRORS :      0

COURSE DESCRIPTOR LOAD PROGRAM IS TERMINATING
```

PROGRAM NAME : P7B

ENTRY POINT : IHECMS (Parameter Supplied by Executive RUNP7B at Entry)

RUN EXECUTIVES(S) : RUNP7B, DEVICE, JCL

USER OUTPUT VIA : Terminal, Offline Printer, or Both

USER PROMPTING : At Executive Level: (1) Output Device Select
(2) Print Option (A, B, or C) Select

FUNCTION : Print Data Contained Within or Associated With the Abbreviated Course File (File Number 7)

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
MPTR	OUTPUT	N/A : THIS IS THE OFFLINE PRINTER			FA, 121, 121
MREF	INPUT	1	DASD	ETAM/MREF	F, 80, 80
COURSE	DIRECT	7	DASD	ETAM/COURSE	F, 80, 80
INDESC	INPUT	3	DASD	ETAM/DIND	F, 80, 80
DCDP	INPUT	9	DASD	ETAM/DCDP	F, 1200, 1200

Discussion

This module serves to provide printer dumps of the contents of the Abbreviated Course file. Due to the size of the output, offline print is recommended as the most economical method.

Listing of EXECUTIVE = RUNP7B

```
&TYPE OFF
&COMMENT G.P. PRINT EXEC FOR COURSES
&COMMENT ENTRY/DEVICE PROMPTS DONE HERE AND IN EXEC = DEVICE
&SPACE
&PRINT ABBR. COURSE FILE PRINT PROGRAM IS STARTING
&SPACE
&PRINT ENTER SINGLE CHARACTER OPTION (OR QUIT TO TERMINATE)
&PRINT A = PRINT ALL COURSES, SORTED BY CIN
&PRINT B = PRINT ALL COURSES IN CDP ORDER
&PRINT C = LIST DESCRIPTORS ASSIGNED TO COURSE DATA
```

```
-STP1 &READ ARGS
      &IF &1 EQ QUIT &QUIT
      &ALPHA0 = &1
      &IF &1 EQ A &GOTO -STP2
      &IF &1 EQ B &GOTO -STP2
      &IF &1 EQ C &GOTO -STP2
      &SPACE
      &PRINT UNABLE TO DETERMINE ENTRY TYPE - TRY AGAIN
      &SPACE
      &GOTO -STP1
-STP2 EXEC DEVICE
      &ALPHA2 = &ALPHA0 !! &ALPHA1
-STP3 &SPACE
EXEC JCL
FILEDEF DIND CLEAR
FILEDEF INDESC DSK ETAM DIND RE F LR 80 BL 80
FILEDEF MPTR PTR RE FA LR 121 BL 121
LOAD P7B (CLEAR LIBE) PLILIB
START IHECMS &ALPHA2 (BRIEF)
FILEDEF * CLEAR
      &EXIT
```

Program P7B Specific User Instructions

As shown in the samples following, the User will be prompted for both output device and type of output. The type of output is categorized by the following options in this program:

Option	Meaning
A	Print the contents of each Abbreviated Course file record; the output is to be ordered by CIN number.
B	Print the same contents; output order is by CDP number.
C	Print the descriptors associated with Courses; the data for this output is retrieved from the Master Descriptor Index file (File Number 3).

NOTE - To print each of the Course CIN numbers and the actual descriptors assigned to each, use Option "C" of Program P20.

Program P7B Error Messages

The following error message can result during Program P7B execution:

```
** ERROR XX OCCURED ; NREC WAS YYYYY **
```

In this message, "XX" is the error number as explained below. The quantity "YYYYY" is the number of the input record on which the error was detected.

XX	Meaning
01	Print Option "B" was being processed. The CDP Directory File indicated that a record was to be accessed from the Abbreviated Course file. The record number of the new record was outside the bounds of the Course file. Note that the maximum number of allowed records appearing in the Course file is maintained in the first count (subscript = 1) of the Master Reference file (File Number 1).
02	Print Option "C" processing. Master Reference file (subscript = 4) contains the maximum usable record in the Descriptor Index (File Number 3) file. An attempt was made to read a record beyond this allowable limit.

Program P7B Sample Run Output - Print Option "A"

The following represents a sample of an actual run of Program P7B using Print Option "A". The appearance of the output resulting from Option "B" would be the same, except for ordering of the individual records. Accordingly, a "B" Option sample will not be shown.

```
            RUNP7B

ABBP. COURSE FILE PRINT PROGRAM IS STARTING

ENTER SINGLE CHARACTER OPTION (OR QUIT TO TERMINATE)
A = PRINT ALL COURSES, SORTED BY CIN
B = PRINT ALL COURSES IN CDP ORDER
C = LIST DESCRIPTORS ASSIGNED TO COURSE DATA

A

OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
ENTER TERM, PRTR, BOTH, OR QUIT

TERM

$$$EXECUTION:

    PAGE :          1

   CIN    CDP     COURSE TITLE     NOBC NEC  RMS  TPC   LEN THRS LHRS ATTR STRX
-------- ---- ---------------- ---- ---- --- ----- --- ---- ---- ---- ----

A 000111 0327 SWO ADV COMMAND       0000 PDB 04350  40  224   16  0.1  0.1
     PRIORITY :          TYPE CRS : C2      SERVICE CDE : 1      COST/AOB :      0.00
        TRAPS : T           STATUS : A         STATUS DTE : 00000   METH-I : L

A 2E0013 0303 TAL PCO/PXO ORI  0000 0000 KBA 03250  12   80    0  0.0  0.0
     PRIORITY : 12       TYPE CRS : C2      SERVICE CDE : 1      COST/AOB :      0.00
        TRAPS :             STATUS : A         STATUS DTE : 62171   METH-I : P

A 2G0014 0143 MINE C/M OFF     9268 0000 CCB 04311  33  120   30  0.0  0.0
     PRIORITY : 25       TYPE CRS : C2      SERVICE CDE : 1      COST/AOB :      0.00
        TRAPS :             STATUS : A         STATUS DTE : 72183   METH-I : L
```

... Output terminated at this point ...

Program P7B Sample Run Output - Print Option "C"

The following is an actual Program P7B run using Print Option "C".

```
RUNP7B

ABBR. COURSE FILE PRINT PROGRAM IS STARTING

ENTER SINGLE CHARACTER OPTION (OR QUIT TO TERMINATE)
A = PRINT ALL COURSES, SORTED BY CIN
B = PRINT ALL COURSES IN CDP ORDER
C = LIST DESCRIPTORS ASSIGNED TO COURSE DATA

C

OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
ENTER TERM, PRTR, BOTH, OR QUIT

TERM

$$EXECUTION:
    PAGE :          1


-- COURSES

::: REFERENCE KNOWLEDGE
01 SYSTEM PURPOSES
02 ORGANIZATIONAL ROLES
03 CONTEXTS OF OPERATION
04 ORGANIZATIONAL RULES
05 OTHER

::: ENABLING KNOWLEDGE
10 OPERATIONAL GOAL CRITERIA
11 NOMENCLATURE, IDENTIFICATION, LOCATION
12 PROCEDURAL DESCRIPTIONS
13 JOB RELEVANT FACTS,RULES
14 OTHER

::: TASK FORMATS
20 PROCEDURE FORMATS
21 DECISION FORMATS
22 CONSTRUCTION FORMATS
23 OTHER

::: GROSS JOB CATEGORIES
30 OPERATIONS
31 MAINTENANCE
32 SERVICE & ADMINISTRATION
33 COMMAND
34 OTHER
```

```
*** OBJECTIVE TASK VARIABLES AS MANIFEST IN THE TRAINING
40 EQUIPMENT & OBJECTS USED: REAL
41 EQUIPMENT & OBJECTS USED: SIMULATED
42 EQUIPMENT & OBJECTS USED: SYMBOLIC
43 ENVIRONMENTS IN WHICH TASK IS TRAINED: REAL
44 ENVIRONMENTS IN WHICH TASK IS TRAINED: SIMULATED
45 ENVIRONMENTS IN WHICH TASK IS TRAINED: SYMBOLIC
46 TOOLS/INSTRUMENTS USED IN TRAINING: REAL
47 TOOLS/INSTRUMENTS USED IN TRAINING: SIMULATED
48 TOOLS/INSTRUMENTS USED IN TRAINING: SYMBOLIC
49 REF/ENABLING INFO IN DOING TASK: APPLIED
50 REF/ENABLING INFO IN DOING TASK: NOT APPLIED
51 CRITERIA OF TASK PERFORMANCE: REAL
52 CRITERIA OF TASK PERFORMANCE: SIMULATED
53 CRITERIA OF TASK PERFORMANCE: SYMBOLIC

*** TASK FUNCTIONS DOMINANT IN TRAINING
60 GOAL PROJECTION
61 SCAN-DETECT
62 IDENTIFY
63 INTERPRET
64 PROCEDURE FOLLOWING
65 DECIDE
66 CONSTRUCT, PLAN
67 TRACK
68 MOTOR PERFORMANCE
69 INTERPERSONAL INTERACTION
70 RECALL TASK-CYCLE INFORMATION
71 RECALL ENABLING INFORMATION
72 ADAPT IMPROVISATIONALLY/IMPROMPTU


    PAGE :          2

73 OTHER

*** STAGE OF LEARNING
80 ORIENTATION, FAMILIARIZATION
81 TASK NOMENCLATURE, IDENTS, LOCATIONS, FACTS, RULES
82 TASK FORMATS AT CONCEPTUAL LEVEL
83 PROCEDURES AT VERBAL LEVEL ONLY
84 TASK COMPONENTS WITH GUIDANCE
85 ENTIRE JOB-TASK PROCEDURALLY AT BARELY ACCEPTABLE MASTERY
86 HIGHLY PROFICIENT IN WORK CONTEXT
87 UNUSUAL TASK CONDITIONS
88 PERFORMANCE AT KEY MAN LEVEL
89 REFRESHER LEARNING
```

PROGRAM NAME : P8

ENTRY POINT : P8

RUN EXECUTIVES(S) : RUNP8

USER OUTPUT VIA : Terminal Only

USER PROMPTING : None

FUNCTION : Makeup Vehicle Abbreviated Data File, Directory, and Descriptor File (File Numbers 15, 16, and 12, Inclusive)

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	Filename/Filetype	Format
MREF	INOUT	1	DASD	ETAM/MREF	F, 80, 80
DIND	DIRECT	3	DASD	ETAM/DIND	F, 80, 80
VDATA	INPUT	14	DASB	ETAM/SVDATA	F, 80, 80
VCDESC	INPUT	11	DASD	ETAM/SVCDESC	F, 80, 80
DVEH	OUTPUT	16	DASD	ETAM/DVEH	F, 750, 750
VDESC	OUTPUT	12	DASD	ETAM/VDESC	F, 213, 213
VEHS	OUTPUT	15	DASD	ETAM/VEHS	F, 80, 80

Discussion

All three files of the Abbreviated Vehicle data base are created by Program P8. This includes the data file (File Number 15), the Vehicle Directory (File Number 16), and the Vehicle Descriptor file (File Number 12).

Listing of EXECUTIVE = RUNP8

```
      &TYPE OFF
      &COMMENT MAKE VEH DATA/DIRECTORY/DESCRIPTOR FILES
      &COMMENT NO ENTRY PARMS
 FILEDEF MREF DSK ETAM MREF RE F LR 80 BL 80
 FILEDEF DIND DSK ETAM DIND RE F LR 80 BL 80
 FILEDEF VDATA DSK ETAM SVDATA RE F LR 80 BL 80
 FILEDEF VCDESC DSK ETAM SVCDESC RE F LR 80 BL 80
 FILEDEF VDESC DSK ETAM VDESC RE F LR 213 BL 213
 FILEDEF VEHS DSK ETAM VEHS RE F LR 80 BL 80
 FILEDEF DVEH DSK ETAM DVEH RE F LR 750 BL 750
 LOAD P8 (CLEAR LIBE) PLILIB
 START (BRIEF)
 FILEDEF * CLEAR
      &EXIT
```

Program P8 Error Messages

Within the data input deck (File Number 11), a given Stock Number can exist only on one card. Since the deck is sorted, duplicates will be signaled by the following message, and then rejected:

```
** DUPLICATE STOCK NUMBER ON PRECEEDING CARD
   RECORD WILL BE IGNORED **
```

NOTE - See sample output (to follow) for an example of this message.

Within the descriptor input deck, a given numeric descriptor can be illegal. Program P8 verifies the existence of each such descriptor through use of the Master Descriptor Index (File Number 3). If an error is detected, the following message will ensue:

```
** ERROR IN VEH -1234567890123- DESCRIPTOR SUBSCRIPT
   999 DOES NOT EXIST **
```

Since both the Vehicle data cards and the Vehicle descriptor cards are input to P8, discrepancies can exist within the Stock Numbers. The following messages, if applicable, will be printed after the descriptor file has reached EOF and the EOF message printed.

```
** THE FOLLOWING DATA FILE VEH NUMBERS HAVE NO
   MATCH IN THE VEH DESCRIPTOR INPUT **

      XXXXXXXXXXXXX
      YYYYYYYYYYYYY
      ...
```

and/or the following

```
** THE FOLLOWING DESCRIPTOR CARD VEH NUMBERS HAVE NO
   MATCH IN THE VEH DATA FILE **

      XXXXXXXXXXXXX
      YYYYYYYYYYYYY
```

Either or both of the above error message sequences will be absent in the event of no detected errors.

Program P8 Specific User Instructions

Both the input Vehicle data file (File Number 14) and the Vehicle descriptor card file (File Number 11) must be sorted into ascending order by Stock Number before run of Program P8.

Note that both of these sorts are accomplished within the executive STARTUP.

Program P8 Sample Run Output

The following represents sections of an actual run of Program P8. Content sufficient to illustrate the output message sequence has been maintained.

```
          RUNP8
$S
INITILIZE OF VEHICLE DATA AND DESCRIPTOR FILES

NUM  STOCK NUMBER   DESIGNATOR  DEVICE NOMENCLATURE
---  -------------  ----------  -------------------

  1  6910LLC000468  12CD-17    MATH, VECTOR ACCELERATION DEMONSTRATOR T/A
  2  6910LLC000469  12CD-18    MATH, NOMOGRAPHIC PROJECTION DEMONSTRATOR T/A
  3  6910LLC000470  12CD-19    MATH, PROJECTION, DEMONSTRATOR TRAINING AID
  4  6910LLC000471  12CD-21    MATH, ORDER OF DIFFERENTIATION, DEM T/A
```

... Output interrupted at this point ...

```
 57  6940005271901  26A-11     ELECTRONICS FUNDAMENTALS UNITS EFU-1

     *** DUPLICATE STOCK NUMBER ON PRECEEDING CARD
         RECORD WILL BE IGNORED ***

 57  6940005271904  26A-11     LABORATORY CIRCUIT ANALYSIS UNIT CAU-1
 58  6940005606627  26A-11     VOLTAGE REGULATOR LAB CKT ANAL UNIT CAU-3
 59  6940005723015  26A-11     RADAR/MICROWAVE SYS CKT ANALYSIS UNIT CAU-5
 60  6940005723018  26A-11     SONAR LAB CIRCUIT ANALYSIS UNIT CAU-6
 61  6940006641435  15Z1-1     APS-T3, ULTRASONIC TRAINER
 62  6940007123047  26A-11     TRANSISTOR LAB CIRCUIT ANALYSIS UNIT CAU-7
 63  6940008506112  26A-11     SYNCHRO LAB CKT ANALYSIS UNIT CAU-9

*** NOW AT EOF ON DATA CARD INPUT ***
    A TOTAL OF     63 INPUT CARDS WERE READ

*** VEHICLE DIRECTORY HAS BEEN SUCCESSFULLY REWRITTEN

*** LOAD OF VEHICLE DESCRIPTOR FILE FOLLOWS ***

STOCK NUMBER   DEVICE DESCRIPTORS
-------------  ------------------

6910LLC000468   03 11 21 32 45 52 61 71 81
6910LLC000469   03 11 21 32 45 52 61 71 81
6910LLC000470   03 11 21 32 45 52 61 71 81
```

... Output interrupted at this point ...

```
...

 6940LLC003443    05  11  21  32  42  52  61  71  81
 6940LLC003445    05  11  21  32  42  52  61  71  81
 6940LLC003447    05  11  21  32  42  52  61  71  81
 6940LLC003448    05  11  21  32  42  52  61  71  81
 6940LLC003449    05  11  21  32  42  52  61  71  81
 6940005271901    05  11  21  32  42  52  61  71  81
 6940005271904    05  11  21  32  42  52  61  71  81
 6940005606627    05  11  21  32  42  52  61  71  81
 6940005723015    05  11  21  32  42  52  61  71  81
 6940005723018    05  11  21  32  42  52  61  71  81
 6940006641035    09  14  21  33  42  55  66  72  83
 6940007123047    05  11  21  32  42  52  61  71  81
 6940008506112    05  11  21  32  42  52  61  71  81

 *** EOF REACHED ON DESCRIPTOR CARD INPUT FILE ***

            TOTAL NUMBER DESCRIPTOR CARDS INPUT :     63
        TOTAL NUMBER UNIQUE VEHICLE STOCK NUMBERS :     63
 TOTAL MATCHES FOUND IN VEHICLE DATA DIRECTORY :     63

 VEHICLE DATA/DIRECTORY/DESCRIPTOR LOAD PROGRAM
     IS NOW TERMINATING
```

PROGRAM NAME : P8A

ENTRY POINT : IHECMS (Runtime Parameter Supplied by Executive RUNP8A at Entry)

RUN EXECUTIVES(S) : RUNP8A

USER OUTPUT VIA : Terminal, Offline Printer, or Both

USER PROMPTING : At Executive Level: (1) Output Device Select
(2) Print Option (A or B) Select

FUNCTION : Print Data Within or Associated With the Abbreviated Vehicle File (File Number 15)

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
SYSPRINT	OUTPUT	N/A : THIS IS THE OFFLINE PRINTER			FA, 121, 121
MREF	INPUT	1	DASD	ETAM/MREF	F, 80, 80
DIND	DIRECT	3	DASD	ETAM/DIND	F, 80, 80
DVEH	INPUT	16	DASD	ETAM/DVEH	F, 750, 750
VEHS	DIRECT	15	DASD	ETAM/VEHS	F, 80, 80

Discussion

The contents of the Abbreviated Vehicle data file are dumped to the online terminal of offline printer (or both) by use of Program P8A.

Listing of EXECUTIVE = RUNP8A

```
          &TYPE OFF
          &COMMENT G.P. PRINT EXEC FOR VEHICLES
          &COMMENT ENTRY/DEVICE PROMPTS DONE HERE AND IN EXEC = DEVICE
          &SPACE
          &PRINT ABBR. VEHICLE FILE PRINT PROGRAM IS STARTING
          &SPACE
          &PRINT ENTER SINGLE CHARACTER OPTION (OR QUIT TO TERMINATE)
          &PRINT A = PRINT ALL VEHICLE RECORDS IN SORTED ORDER
          &PRINT B = LIST DESCRIPTORS ASSIGNED TO VEHICLE DATA
    -STP1 &READ ARGS
          &IF &1 EQ QUIT &QUIT
          &ALPHA0 = &1
          &IF &1 EQ A &GOTO -STP2
          &IF &1 EQ B &GOTO -STP2
          &SPACE
          &PRINT UNABLE TO DETERMINE ENTRY TYPE - TRY AGAIN
          &SPACE
          &GOTO -STP1
    -STP2 EXEC DEVICE
          &ALPHA2 = &ALPHA0 !! &ALPHA1
    -STP3 &SPACE
    EXEC JCL
    FILEDEF SYSPRINT PTR RE FA LR 121 BL 121
    LOAD P8A (CLEAR LIBE) PLILIB
    START IHECMS &ALPHA2 (BRIEF)
    FILEDEF * CLEAR
          &EXIT
```

Program P8A Specific User Instructions

As shown in the samples to follow, the User will be prompted for both output device and type of output. The type of output is categorized by the following options in this program:

Option	Meaning
A	Print the contents of each Abbreviated Vehicle file record. The order of output will be the natural order of the file, i.e., Vehicle Stock Number.
B	Print the descriptors associated with Vehicles. The data for this output is retrieved from the Descriptor Master Index file (File Number 3).

NOTE - To print each of the Vehicle Stock Numbers and the actual descriptors assigned to each unique Stock Number, use the "V" Option of Program P20.

Program P8A Error Messages

The following error message results during Program P8A execution only in the event of a serious logic defect in the program:

```
** ERROR XX OCCURED ; NREC WAS YYYYY **
```

In the message, "XX" is the error number as explained below. The quantity "YYYYY" indicates the number of the input record on which the error was detected.

XX	Meaning
01	While processing Print Option "B", an attempt was made to access a record that was outside the bounds of the Descriptor Master Index file (ETAM/DIND - File Number 3). Note that the maximum number of records present in File Number 3 is indicated in the fourth (subscript = 4) count in the Master Reference File ETAM/MREF.
02	During Print Option "A" processing, an attempt was made to access a record outside the bounds of the Vehicle data file. Master Reference file (subscript = 7) indicates the maximum record number used in the Vehicle data file (File Number 15).

Program P8A Sample Run Output - Print Option "A"

The following listing is a portion of an actual run of Program P8 using Print Option "A".

```
          RUNP8A

    ABBR. VEHICLE FILE PRINT PROGRAM IS STARTING

    ENTER SINGLE CHARACTER OPTION (OR QUIT TO TERMINATE)
    A = PRINT ALL VEHICLE RECORDS IN SORTED ORDER
    B = LIST DESCRIPTORS ASSIGNED TO VEHICLE DATA

    A

    OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
    ENTER TERM, PRTR, BOTH, OR QUIT

    TERM

    $$$$EXECUTION:

    PAGE :     1

    ITEM    STOCK NUM      DESIGNATOR   DEVICE COST
    ----  -------------  ----------  -------------

       1  6910LLC000468   12CD-17           $50.00
              MATH, VECTOR ACCELERATION DEMONSTRATOR T/A

       2  6910LLC000469   12CD-18        $2,000.00
              MATH, NOMOGRAPHIC PROJECTION DEMONSTRATOR T/A

       3  6910LLC000470   12CD-19        $1,500.00
              MATH, PROJECTION, DEMONSTRATOR TRAINING AID

       4  6910LLC000471   12CD-21        $1,000.00
              MATH, ORDER OF DIFFERENTIATION, DEM T/A

       5  6910LLC000472   12CD-22           $10.00
              MATH, FINITE ROTATION, DEMONSTRATOR T/A

       6  6910LLC000473   12CD-24           $50.00
              MATH, DIRECTION COSINE, DEMONSTRATOR T/A

       7  6910LLC000474   12CD-25        $1,500.00
              MATH, SKEW LINE, DEMONSTRATOR TRAINING AID

       8  6910LLC000900   3E34-1         $1,020.00
              106MM RECOILLESS RIFLE, FIRING MECH. TRANSP.
```

... Output terminated at this point ...

Program P8A Sample Run Output - Print Option "B"

Using Print Option "B", the following represents a sample of the resulting output of Program P8A.

```
          RUNP8A

ABBR. VEHICLE FILE PRINT PROGRAM IS STARTING

ENTER SINGLE CHARACTER OPTION (OR QUIT TO TERMINATE)
A = PRINT ALL VEHICLE RECORDS IN SORTED ORDER
B = LIST DESCRIPTORS ASSIGNED TO VEHICLE DATA

B

OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
ENTER TERM, PRTR, BOTH, OR QUIT

TERM

$$EXECUTION:

PAGE :     1


-- VEHICLES

::: VEHICLE TYPES
01 INSTRUCTOR
02 STATIC GRAPHICS
03 ANIMATED GRAPHICS
04 AUDIO
05 PHYSICAL MODELS
06 PROCEDURAL TRAINERS: SYMBOLIC
07 PROCEDURAL TRAINERS: PHYSICAL BUT NON-FUNCTIONAL
08 PROCEDURAL TRAINERS: FUNCTIONAL
09 TASK & SYSTEM SIMULATORS
10 REAL EQUIPMENT ITSELF

::: CLASS OF TRAINING OBJECTIVE
11 REFERENCE KNOWLEDGE
12 KNOWLEDGE, TASK SPECIFIC/ENABLING
13 TASK-SKILL FORMATS
14 SKILL TRAINING

::: VEHICLE PROPERTIES
21 VISUAL
22 AUDITORY
23 KINESTHETIC/VESTIBULAR
24 TACTILE
```

::: TYPE OF CONTENT DISPLAYED
31 TEXT-VERBAL
32 DIAGRAMMATIC
33 ABSTRACTED PICTORIAL REPRESENTATION
34 PICTORIAL REPRESENTATIONS
35 PHYSICAL REPRESENTATIONS
36 OTHER

::: TYPE OF PRESENTATIONAL SEQUENCE
41 LIBRARY OF FRAMES OR ITEMS
42 PRESENTATION SEQUENCE NOT APPLICABLE
43 FIXED SEQUENTIAL FRAMES OR ITEMS
44 RANDOM SELECTION OF FRAME SEQUENCES
45 DYNAMIC CHANGE OF CONTENT WITHIN FRAME

::: SELECTION SOURCE FOR SEQUENCING
51 INTERNAL PROGRAM
52 INSTRUCTOR
53 STUDENT CHOICE
54 STUDENT PERFORMANCE
55 COMBINATIONS OF THE ABOVE

::: TYPE OF EXTERNAL CONTROL OPERATED BY STUDENT
61 NOT APPLICABLE DIRECTLY
62 ARTIFICIAL OR SYMBOLIC RESPONSE
63 REPRESENTATIONAL RESPONSE BY SYMBOLIC SELECTION
64 REPRESENTATIONAL RESPONSE BY DUMMY CONTROL ACTIVATION
65 TASK-MANIPULATIVE RESPONSE, NON-DYNAMIC IN TIME AND FORCE

66 TASK-MANIPULATIVE RESPONSE, DYNAMIC AND INTERACTIVE

::: FEEDBACK PRESENTATION LOGIC
71 NOT APPLICABLE
72 SELECTS NEXT STIMULUS ITEM OR SEQUENCE
73 GIVES EVALUATION OF PRECEDING RESPONSE
74 SELECTS AND PRESENTS GUIDANCE INFO

::: RESPONSE EVALUATION LOGIC
81 NOT INTERNAL-DEPENDS ON INSTRUCTOR OR STUDENT EVALUATION
82 EVALUATION LIMITED TO STUDENT'S IMMEDIATE RESPONSE
83 EVALUATION EXTENDED TO A SET OF STUDENT RESPONSES
84 TOLERANCE LIMITS ON ACCEPTABLE STUDENT RESPONSE: FIXED
85 TOLERANCE LIMITS ON ACCEPTABLE STUDENT RESPONSE: VARIABLE

PROGRAM NAME : P9

ENTRY POINT : P9

RUN EXECUTIVES(S) : RUNP9

USER OUTPUT VIA : Terminal, Offline Printer, or Both

USER PROMPTING : Program Level : Output Device Select

FUNCTION : Makeup Task Abbreviated Data File, Directory, and Descriptor File (File Numbers 22, 23, and 19, incl.)

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
SYSPRINT	OUTPUT	N/A : THIS IS THE OFFLINE PRINTER			FA, 121, 121
MREF	INOUT	1	DASD	ETAM/MREF	F, 80, 80
DIND	DIRECT	3	DASD	ETAM/DIND	F, 80, 80
TDATA	INPUT	21	DASD	ETAM/STDATA	F, 80, 80
TCDESC	INPUT	18	DASD	ETAM/STDSCR	F, 80, 80
DTASK	OUTPUT	23	DASD	ETAM/DTASK	F, 1500, 1500
TASKS	OUTPUT	22	DASD	ETAM/TASKS	F, 80, 80
TDESC	OUTPUT	19	DASD	ETAM/TDESC	F,213, 213

Discussion

Program P9 serves to generate all three files of the Abbreviated Tasks data base. This includes the data file (File Number 22), the Tasks Directory (File Number 23), and the Tasks Descriptor file (File Number 19).

Listing of EXECUTIVE = RUNP9

```
      &TYPE OFF
      &COMMENT LOAD TASK DATA/DIRECTORY/DESCRIPTORS
      &COMMENT NO EXEC ENTRY PARMS
      &COMMENT PROGRAM P9 WILL PROMPT FOR OUTPUT DEVICE,
      &COMMENT TERMINAL, OFFLINE PRINTER, BOTH, QUIT.
 FILEDEF MREF DSK ETAM MREF RE F LR 80
 FILEDEF DIND DSK ETAM DIND RE F LR 80 BL 80
 FILEDEF TDATA DSK ETAM STDATA RE F LR 80 BL 80
 FILEDEF TCDESC DSK ETAM STDSCR RE F LR 80 BL 80
 FILEDEF DTASK DSK ETAM DTASK RE F LR 1500 BL 1500
 FILEDEF TASKS DSK ETAM TASKS RE F LR 80 BL 80
 FILEDEF TDESC DSK ETAM TDESC RE F LR 213 BL 213
 FILEDEF SYSPRINT PTR RE FA LR 121 BL 121
 LOAD P9 (CLEAR LIBE) PLILIB
 START (BRIEF)
 FILEDEF * CLEAR
      &EXIT
```

Program P9 Error Messages

Within the data input card deck (File Number 21), a given Rate/Rank/Jobtask Number cannot be duplicated on more than one card. Since this file is sorted by the contents of these data fields, duplicates can be detected by Program P9. In such a case, the following error message text will be output:

```
** DUPLICATE STOCK NUMBER ON PRECEEDING CARD RECORD
   WILL BE IGNORED **
```

Program P9 verifies the legality of each of the descriptors in the descriptor input file (File Number 18). The Master Descriptor Index (File Number 3) is used for this purpose. If an error is detected, the following message is output:

```
** ERROR IN TASK NUMBER -XXX   YY  ZZZZZ- DESCRIPTOR
   SUBSCRIPT SSS DOES NOT EXIST **
```

where "XXX" is the Task Rate, "YY" the Rank, and "ZZZZZ" the Jobtask number.

Since both the Task data cards and descriptor cards are input to Program P9, it is possible for a discrepancy to exist within the key fields of each (first 13 characters of each card - includes Rate, Rank, and Jobtask number). If such discrepancies are detected, the occurance is saved within the Program and the messages illustrated below will be printed following the message signifying end-of-file on the descriptor card input.

```
** THE FOLLOWING DATA FILE TASK NUMBERS
   HAVE NO MATCH IN THE DESCRIPTOR CARD FILE **

         XXX    XXXXXXXXXXXXX
         XXX    XXXXXXXXXXXXX
                ... Etc.
```

and/or the following

```
** THE FOLLOWING DESCRIPTOR CARD TASK NUMBERS
   HAVE NO MATCH IN THE DATA FILE **

         YYY    YYYYYYYYYYYYY
         YYY    YYYYYYYYYYYYY
```

Either or both of the above error message sequences will be absent in the event of no detected errors.

<u>Program P9 Specific User Instructions</u>

Both input files - the data file (File Number 21) and the descriptor input file (File Number 18) must be sorted into ascending order by the first 13 characters of each record <u>before</u> run of Program P9.

Note that both of the required sorts are accomplished within the executive STARTUP.

Program P9 Sample Run Output

The following represents sections of an actual run of Program P9. Content sufficient to illustrate the output message sequence has been maintained.

```
          RUNP9

$$$
TASK DATA/DIRECTORY/DESCRIPTOR LOAD PROGRAM IS NOW STARTING

OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, OR BOTH?
ENTER TERM, PRTR, BOTH, OR QUIT
_TERM

      ** LOAD OF TASK ABBREVIATED DATA FILE FOLLOWS

NUMBER     RATE     RANK     TASK     TASK TITLE

     1      AN              20350     AIRCRAFT UNDER EMERGENCY CONDITIONS
     2      AN              20351     IDENTIFY MARKINGS INDICATING DANGEROUS AREAS
     3      AN              20352     IDENTIFY TOXIC PROPERTIES OF CLEANING
                                      MATERIALS
     4      AN              20368     RECOGNIZE ARMED EJECTION SEATS
     5      AN              24425     HANDLE AIRCRAFT BATTERIES
     6      AN              25352     KNOW NAVAL AIR ARM GENERAL ORGANIZATION
     7      AN              25353     IDENTIFY NAMES, FUNCTIONS OF NAVAL AIR GROUPS
     8      AN              25354     IDENTIFY NAVAL AIRCRAFT SQUADRONS
     9      AN              42353     STAND AIRCRAFT SECURITY WATCH
```

... Output interrupted at this point ...

The above text will be followed by:

```
** EOF ON DATA INPUT FILE
   CARDS PROCESSED : XX,XXX **

** LOAD OF TASK DESCRIPTOR CARDS FOLLOWS **
```

(Followed by print of Task Descriptor Cards)

```
** EOF REACHED ON TASK DESCRIPTOR INPUT FILE
   CARDS PROCESSED : XX,XXX **
```

PROGRAM NAME : P9A

ENTRY POINT : IHECMS (Execution Parameter Supplied by Executive RUNP9A at Entry)

RUN EXECUTIVES(S) : RUNP9A

USER OUTPUT VIA : Terminal, Offline Printer, or Both

USER PROMPTING : At Executive Level: (1) Output Device Select
(2) Print Option Select

FUNCTION : Print Data Within or Associated with the Abbreviated Tasks File (File Number 22)

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
MREF	INPUT	1	DASD	ETAM/MREF	F, 80, 80
DIND	DIRECT	3	DASD	ETAM/DIND	F, 80, 80
DTASK	INPUT	23	DASD	ETAM/DTASK	F, 1500, 1500
TASKS	DIRECT	22	DASD	ETAM/TASKS	F, 80, 80
SYSPRINT	OUTPUT	N/A : THIS IS THE OFFLINE PRINTER			FA, 121, 121

Discussion

Program P9A to printer dump the contents of the Abbreviated Tasks data file (File Number 22). At User option, the dump may be directed to either (or both) the online terminal or the offline printer.

Listing of EXECUTIVE = RUNP9A

```
&TYPE OFF
&COMMENT G.P. PRINT EXEC FOR TASKS
&COMMENT ENTRY/DEVICE PROMPTS DONE HERE AND IN EXEC = DEVICE
&SPACE
&PRINT ABBR. TASKS FILE PRINT PROGRAM IS STARTING
&SPACE
&PRINT ENTER SINGLE CHARACTER OPTION (OR QUIT TO TERMINATE)
&PRINT A = PRINT ALL TASK RECORDS IN SORTED ORDER
&PRINT B = LIST DESCRIPTORS ASSIGNED TO TASK DATA
```

```
-STP1 &READ ARGS
      &IF &1 EQ QUIT &QUIT
      &ALPHA0 = &1
      &IF &1 EQ A &GOTO -STP2
      &IF &1 EQ B &GOTO -STP2
      &SPACE
      &PRINT UNABLE TO DETERMINE ENTRY TYPE - TRY AGAIN
      &SPACE
      &GOTO -STP1
-STP2 EXEC DEVICE
      &ALPHA2 = &ALPHA0 !! &ALPHA1
-STP3 &SPACE
EXEC JCL
FILEDEF SYSPRINT PTR RE FA LR 121 BL 121
LOAD P9A (CLEAR LIBE) PLILIB
START IHECMS &ALPHA2 (BRIEF)
FILEDEF * CLEAR
      &EXIT
```

Program P9A Specific User Instructions

As shown in the samples to follow, the User is prompted to select the output device and the type of output. Within Program P9A, the following output options are available:

Option	Meaning
A	Print the contents of the Abbreviated Tasks file. The order of the output will be the natural order of the file, i.e., by the contents of the 13-character field containing the concatenation of Rate, Rank, and Jobtask Number.
B	Print the descriptors associated with Tasks. The data for this output is retrieved from the Descriptor Master Index file (File Number 3). NOTE - To command the print of the actual descriptors assigned to each unique task, use the "T" Option of Program P20.

Program P9A Error Messages

The following error text can be generated by Program P9A:

```
** ERROR XX OCCURED ; NREC WAS YYYYY **
```

In the above message, "XX" is the error number as explained below. The quantity "YYYYY" indicates the number of the input record that was being processed when the error was detected.

XX	Meaning
01	During processing of Print Option "B", an attempt was made to read a record that was outside the bounds of the Descriptor Master Index file (ETAM/DIND - File Number 3). The Master Reference file (ETAM/MREF - File Number 1) indicates the maximum number of records available in the ETAM/DIND file. This count is available in the fourth (subscript = 4) count of MREF.
02	While processing Print Option "A", an attempt was made to access a record outside the bounds of the Tasks data file (File Number 22). The Master Reference file (subscript = 11) indicates the maximum record number that can be accessed in the Tasks data file.

Program P9A Sample Run Output - Print Option "A"

The following listing is an example of the output that results from selection of Print Option "A".

```
          RUNP9A


     ABBR. TASKS FILE PRINT PROGRAM IS STARTING

     ENTER SINGLE CHARACTER OPTION (OR QUIT TO TERMINATE)
     A = PRINT ALL TASK RECORDS IN SORTED ORDER
     B = LIST DESCRIPTORS ASSIGNED TO TASK DATA

     A

     OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
     ENTER TERM, PRTR, BOTH, OR QUIT
     TERM

     $$EXECUTION:

     PAGE :     1

     ITEM   RATE  RANK  TASK    COST-PER-BILLET
     ----   ----- ----  ------  ---------------


       1    AN          20350            $0.00
               AIRCRAFT UNDER EMERGENCY CONDITIONS

       2    AN          20351            $0.00
               IDENTIFY MARKINGS INDICATING DANGEROUS AREAS

       3    AN          20352            $0.00
               IDENTIFY TOXIC PROPERTIES OF CLEANING MATERIALS

       4    AN          20368            $0.00
               RECOGNIZE ARMED EJECTION SEATS

       5    AN          24425            $0.00
               HANDLE AIRCRAFT BATTERIES

       6    AN          25352            $0.00
               KNOW NAVAL AIR ARM GENERAL ORGANIZATION

       7    AN          25353            $0.00
               IDENTIFY NAMES, FUNCTIONS OF NAVAL AIR GROUPS
```

... Output terminated at this point ...

Program P9A Sample Run Output - Print Option "B"

The following is a sample of the output (complete) from Program P9A that results from Print Option "B" selection.

```
          RUNP9A

ABBR. TASKS FILE PRINT PROGRAM IS STARTING

ENTER SINGLE CHARACTER OPTION (OR QUIT TO TERMINATE)
A = PRINT ALL TASK RECORDS IN SORTED ORDER
B = LIST DESCRIPTORS ASSIGNED TO TASK DATA
B

OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
ENTER TERM, PRTR, BOTH, OR QUIT
TERM

$EXECUTION:

PAGE :     1


-- TASKS

** ADMINISTRATIVE ROUTINE PAPERWORK
01 FORMS FILLING
02 DOCUMENT-FILE MANAGEMENT
03 DECODE-ENCODE
04 SCREEN-FILTER DISTRIBUTE
05 OTHER

** ADMINISTRATIVE NON-ROUTINE PAPERWORK
11 CONSTRUCT MESSAGES-REPORTS
12 ANALYZE-INTERPRET
13 CONSTRUCT RECOMMENDATION-PROPOSAL
14 CONSTRUCT PLAN
15 OTHER

** ADMINISTRATIVE OFFICE EQUIPMENT OPERATION
21 TYPEWRITER
22 TELEPHONE,ETC.
23 REPRODUCER
24 COMPUTER TERMINAL
25 TELETYPE
26 OTHER
```

```
** INTERPERSONAL: INFORM-INSTRUCT-MANAGE
31 BRIEF-DEBRIEF
32 INSTRUCT-TRAIN
33 ASSIGN, MONITOR, COORDINATE
34 EVALUATE
35 OTHER

** TECHNICAL PROCEDURES
41 SEQUENTIAL
42 STRATEGIC, ADAPTIVE
43 INTERPERSONAL,TEAM MEMBER
44 OTHER

** TECHNICAL TYPE OF PROCEDURE(MAIN EMPHASIS)
51 SCAN-DETECT: SYMBOLIC (INCL. MAPS,RADAR,TRANSDUCED SIGNALS)
52 SCAN-DETECT: NATURAL
53 IDENTIFY: SYMBOLIC(INCL. TRANSDUCED SIGNALS)
54 IDENTIFY: NATURAL
55 INTERPRET:SYMBOLIC
56 INTERPRET: NATURAL
57 PERCEPTUAL-MOTOR
58 COGNITIVE OPERATIONS
59 MANUAL
60 COMMUNICATE

** TECHNICAL WITH OR WITHOUT EQUIPMENT
61 WITH EQUIPMENT (PAPER IS EQUIPMENT)
62 WITHOUT EQUIPMENT

** TECHNICAL DECIDE
71 DIAGNOSE-ANALYZE




PAGE :     2

72 SELECT-CHOOSE
73 UNDER STRESS OR LOAD
74 OTHER

** TECHNICAL CONSTRUCT-REPAIR, PLAN
81 MANUAL CONSTRUCT OR REPAIR
82 COGNITIVE CONSTRUCT, PLAN

** TECHNICAL TRACK-AIM-STEER
91 APPLICABLE
92 TIME STRESS
93 INFORMATION-LOAD STRESS
```

PROGRAM NAME : P14

ENTRY POINT : P14

RUN EXECUTIVES(S) : RUNP14

USER OUTPUT VIA : Terminal Only

USER PROMPTING : Executive Level Only

FUNCTION : Extract Abbreviated Course File Data from the NITRAS MCRF Extract Tape

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
NITRAS	INPUT	24	Tape	Not Applicable	FB, 810, 3240
DESCR	INPUT	5	DASD	ETAM/SCDESC	Card Image
COURSE	OUTPUT	7	DASD	ETAM/COURSE	Card Image

Discussion

This program constructs the Abbreviated Course data base. Using the CIN numbers as indicated on the input descriptor cards (File Number 5, above), matching data is sought on the NITRAS MCRF extract tape. If a match occurs, the data is selected from the NITRAS tape.

If a CIN number remains unmatched on a descriptor card, a message is output to indicate the missing data. Samples of this condition follow.

Listing of EXECUTIVE = RUNP14

```
      &TYPE OFF
      &COMMENT MAKEUP OF ABBR. COURSE DATA BASE.
      &COMMENT NO ENTRY PARMS.
      &COMMENT NITRAS TAPE MUST BE MOUNTED PRIOR TO INVOCATION
      &COMMENT OF THIS EXEC.
      &SPACE
      &PRINT IS NITRAS TAPE ALREADY MOUNTED AND MOUNT VERIFIED?
-STP1 &PRINT RESPOND YES OR NO
      &SPACE
      &READ ARGS
      &IF &1 EQ YES &GOTO -YES
      &IF &1 EQ NO &GOTO -NO
      &PRINT UNABLE TO DETERMINE LAST ENTRY TYPE - TRY AGAIN
      &SPACE
      &GOTO -STP1
-NO   &PRINT COMPLETE PROCEDURE FOR NITRAS TAPE MOUNT, AND
      &PRINT INVOKE EXECUTIVE RUNP14 AGAIN.
-FIN  &SPACE
      &PRINT EXECUTIVE RUNP14 IS NOW TERMINATING
-OUT  &SPACE
      &QUIT
-YES  &SPACE
TAPEIO FSF TAP2
      &IF &INDEX0 EQ 0 &GOTO -RUNIT
      &PRINT TAPE FORWARD FILE SKIP BROKE; RETURN CODE WAS &INDEX0
      &GOTO -FIN
-RUNIT FILEDEF NITRAS TAP2 RE FB LR 810 BL 3240 DEN 3
FILEDEF DESCR DSK ETAM SCDESC RE F LR 80 BL 80
FILEDEF COURSE DSK ETAM COURSE2 RE F LR 80 BL 80
LOAD P14 (CLEAR LIBE) PLILIB
START (BRIEF)
FILEDEF * CLEAR
      &EXIT
```

Program P14 Error Messages

The only error of significance in this module results after reaching an end-of-file on the NITRAS tape, when more (one/more) descriptor cards remain to be read. An instance of this is shown in the sample output below. The error message follows the normal EOF ON NITRAS-TAPE INPUT message for CIN number 83000010.

Program P14 Sample Run Output

The following represents an execution of Program P14. Only a cut of the output is shown in the interest of brevity. For a listing of the contents of the actual Course Abbreviated data base, see the output listings that have resulted from Program P7B output.

```
RUNP14


$$
ABBREVIATED COURSE DATA BASE EXTRACT PROGRAM
       DATE : 04/18/77
       TIME : 1350

DESCRIPTOR    CIN            NITRAS        COURSE
 CARD NUM    COUNT    CIN    REC NUM        CDP       COURSE SHORT TITLE

      1          1  A 000111      13        0327       SWO ADV COMMAND

      2          2  A1010108     213        516U       WRA-4 CMB MA
                                 214        5535       WRA-4 CMB /FINIS
                                 215        7654       WRA-4 CMB MA

      3 ***      3  A1020060   - NO CIN MATCH IN NITRAS

      4 ***      4  A1020093   - NO CIN MATCH IN NITRAS

      5 ***      5  A1020095   - NO CIN MATCH IN NITRAS

      6          6  A1930050      794  2213     INERT NAV PRINC

      7          7  A 2E0013      881  0303     TAL PCO/PXO ORI

      ETC ...

** EOF ON NITRAS-TAPE INPUT **

    132 ***   121  83000010   - NO CIN MATCH IN NITRAS -


*   ** EOF ON CARD-DESCRIPTOR INPUT **

      132  :  TOTAL DESCRIPTOR CARDS READ
      121  :  TOTAL CIN NUMBERS INPUT
       50  :  TOTAL CIN ERRORS (NO MATCH IN NITRAS)
      111  :  TOTAL COURSES ECTRACTED
    4,127  :  LAST RECORD READ IN NITRAS
```

Program P14 Specific User Instructions

In regards the NITRAS MCRF Extract Tape:

* The tape must be in the format shown in the File Section for File Number 24.

* The tape must be sorted into ascending order by CIN number, by CDP number. If an IBM sort package is used for this purpose on the 360/370 Series IBM computers, the following OS SORT control card is required:

```
SORT FIELDS=(5,8,CH,A,1,4,CH,A),FILSZ=E5000
```

* When the tape has been posted to the NCSS time-sharing vendor's computer site, have the local NCSS Representative obtain a NCSS tape library "Bin Number" for the tape.

* When ready to process the tape, precede the RUNP14 executive command by the following online terminal entry:

```
MOUNT TAPE binnumber AS TAP2 RINGOUT
```

  Wait for a response from the NCSS I/O Operator. before proceeding with execution of RUNP14.

* RUNP14 executive presumes a Standard-Label OS tape; a forward-skip file is executed to pass over the volume identification and data set label. This is accomplished in RUNP14 EXEC line:

```
TAPEIO FSF TAP2
```

  If a non-labeled tape is to be processed, a change must be made to the executive.

PROGRAM NAME : P16

ENTRY POINT : P16

RUN EXECUTIVES(S) : RUNP16

USER OUTPUT VIA : Terminal Only

USER PROMPTING : None

FUNCTION : Construct Descriptor Master Index File (File Number 3)

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
MREF	INOUT	1	DASD	ETAM/MREF	F, 80, 80
DIND	OUTPUT	3	DASD	ETAM/DIND	F, 80, 80
CARDS	INPUT	2	DASD	ETAM/DICRDS	F, 80, 80

Discussion

The Descriptor Master Index (File Number 3) contains card-image records in the following sequence:

Record Number	Contents
0	Header record, containing 40 binary halfword counts. Count(1) is the zero-origin maximum record number in the file (last record position used).
1 - 4	Course index contents; these cards will be read into the PL/1 structure called DINDEX.
5 - 8	Vehicle index contents; same comments as above.
9 - 12	Task index contents; same comments as above.
13 - N1	Card records as read from the input file (File Number 2) descriptive of Course type descriptors.
(N1+1) - N2	Vehicle descriptor cards.
(N2+1) - N3	Task descriptor cards; end-of-file follows.

Listing of EXECUTIVE = RUNP16

```
     &TYPE OFF
     &COMMENT MAKEUP OF MASTER DESCRIPTOR INDEX (DIND)
     &COMMENT NO ENTRY PARMS
FILEDEF CARDS DSK ETAM DICRDS RE F LR 80 BL 80
FILEDEF MREF DSK ETAM MREF RE F LR 80 BL 80
FILEDEF DIND DSK ETAM DIND RE F LR 80 BL 80
LOAD P16 (CLEAR LIBE) PLILIB
START (BRIEF)
FILEDEF CARDS CLEAR
FILEDEF MREF CLEAR
FILEDEF DIND CLEAR
     &EXIT
```

Program P16 Error Messages

The following general error message sequence can result from errors detected in the run of P16:

> ** ERROR NUMBER X HAS OCCURED **
> (Followed by descriptive line)
>
> (Followed by the two standard terminus messages - as shown on the sample run)

In the message first line, X has the following significance; X also determines the exact contents of the message second line.

Number X	Message Second Line
1	FIRST CARD NOT (--) CARD
2	DUPLICATE (--) CARD
3	TYPE OF (--) CARD INVALID
4	TYPE (**) CARD MUST FOLLOW TYPE (--)
5	CATEGORY HAS NO DESCRIPTORS
6	DESCRIPTOR NUMBER CONVERSION ERROR
7	CATEGORY TABLE SIZE EXCEEDED, MAX = 15

Program P16 Sample Error Message

```
** ERROR NUMBER   2 HAS OCCURED **
DUPLICATE (--) CARD

** TOTAL DESCRIPTOR INDEX CARDS READ : 44

MASTER DESCRIPTOR INDEX PROCESSING IS NOW TERMINATING
```

Program P16 Sample Run Output

The following represents an actual run of P16. The output is shown in its entirety; this represents the contents of the Descriptor Index in the current ETAM implementation.

```
           RUNP16

   $$$$$
   MASTER DESCRIPTOR INDEX CONSTRUCT PROGRAM IS NOW STARTING

   DESCRIPTOR CARD FILE LISTING FOLLOWS
   -------------------------------------



   -- COURSES

   ::: REFERENCE KNOWLEDGE
   01 SYSTEM PURPOSES
   02 ORGANIZATIONAL ROLES
   03 CONTEXTS OF OPERATION
   04 ORGANIZATIONAL RULES
   05 OTHER

   ::: ENABLING KNOWLEDGE
   10 OPERATIONAL GOAL CRITERIA
   11 NOMENCLATURE, IDENTIFICATION, LOCATION
   12 PROCEDURAL DESCRIPTIONS
   13 JOB RELEVANT FACTS,RULES
   14 OTHER

   ::: TASK FORMATS
   20 PROCEDURE FORMATS
   21 DECISION FORMATS
   22 CONSTRUCTION FORMATS
   23 OTHER

   ::: CROSS JOB CATEGORIES
   30 OPERATIONS
   31 MAINTENANCE
   32 SERVICE & ADMINISTRATION
   33 COMMAND
   34 OTHER
```

```
::: OBJECTIVE TASK VARIABLES AS MANIFEST IN THE TRAINING
40 EQUIPMENT & OBJECTS USED: REAL
41 EQUIPMENT & OBJECTS USED: SIMULATED
42 EQUIPMENT & OBJECTS USED: SYMBOLIC
43 ENVIRONMENTS IN WHICH TASK IS TRAINED: REAL
44 ENVIRONMENTS IN WHICH TASK IS TRAINED: SIMULATED
45 ENVIRONMENTS IN WHICH TASK IS TRAINED: SYMBOLIC
46 TOOLS/INSTRUMENTS USED IN TRAINING: REAL
47 TOOLS/INSTRUMENTS USED IN TRAINING: SIMULATED
48 TOOLS/INSTRUMENTS USED IN TRAINING: SYMBOLIC
49 REF/ENABLING INFO IN DOING TASK: APPLIED
50 REF/ENABLING INFO IN DOING TASK: NOT APPLIED
51 CRITERIA OF TASK PERFORMANCE: REAL
52 CRITERIA OF TASK PERFORMANCE: SIMULATED
53 CRITERIA OF TASK PERFORMANCE: SYMBOLIC

::: TASK FUNCTIONS DOMINANT IN TRAINING
60 GOAL PROJECTION
61 SCAN-DETECT
62 IDENTIFY
63 INTERPRET
64 PROCEDURE FOLLOWING
65 DECIDE
66 CONSTRUCT, PLAN
67 TRACK
68 MOTOR PERFORMANCE
69 INTERPERSONAL INTERACTION
70 RECALL TASK-CYCLE INFORMATION
71 RECALL ENABLING INFORMATION
72 ADAPT IMPROVISATIONALLY/IMPROMPTU
73 OTHER

::: STAGE OF LEARNING
80 ORIENTATION, FAMILIARIZATION
81 TASK NOMENCLATURE, IDENTS, LOCATIONS, FACTS, RULES
82 TASK FORMATS AT CONCEPTUAL LEVEL
83 PROCEDURES AT VERBAL LEVEL ONLY
84 TASK COMPONENTS WITH GUIDANCE
85 ENTIRE JOB-TASK PROCEDURALLY AT BARELY ACCEPTABLE MASTERY
86 HIGHLY PROFICIENT IN WORK CONTEXT
87 UNUSUAL TASK CONDITIONS
88 PERFORMANCE AT KEY MAN LEVEL
89 REFRESHER LEARNING


-- VEHICLES

::: VEHICLE TYPES
01 INSTRUCTOR
02 STATIC GRAPHICS
```

```
03 ANIMATED GRAPHICS
04 AUDIO
05 PHYSICAL MODELS
06 PROCEDURAL TRAINERS: SYMBOLIC
07 PROCEDURAL TRAINERS: PHYSICAL BUT NON-FUNCTIONAL
08 PROCEDURAL TRAINERS: FUNCTIONAL
09 TASK & SYSTEM SIMULATORS
10 REAL EQUIPMENT ITSELF

::: CLASS OF TRAINING OBJECTIVE
11 REFERENCE KNOWLEDGE
12 KNOWLEDGE, TASK SPECIFIC/ENABLING
13 TASK-SKILL FORMATS
14 SKILL TRAINING

::: VEHICLE PROPERTIES
21 VISUAL
22 AUDITORY
23 KINESTHETIC/VESTIBULAR
24 TACTILE

::: TYPE OF CONTENT DISPLAYED
31 TEXT-VERBAL
32 DIAGRAMMATIC
33 ABSTRACTED PICTORIAL REPRESENTATION
34 PICTORIAL REPRESENTATIONS
35 PHYSICAL REPRESENTATIONS
36 OTHER

::: TYPE OF PRESENTATIONAL SEQUENCE
41 LIBRARY OF FRAMES OR ITEMS
42 PRESENTATION SEQUENCE NOT APPLICABLE
43 FIXED SEQUENTIAL FRAMES OR ITEMS
44 RANDOM SELECTION OF FRAME SEQUENCES
45 DYNAMIC CHANGE OF CONTENT WITHIN FRAME

::: SELECTION SOURCE FOR SEQUENCING
51 INTERNAL PROGRAM
52 INSTRUCTOR
53 STUDENT CHOICE
54 STUDENT PERFORMANCE
55 COMBINATIONS OF THE ABOVE

::: TYPE OF EXTERNAL CONTROL OPERATED BY STUDENT
61 NOT APPLICABLE DIRECTLY
62 ARTIFICIAL OR SYMBOLIC RESPONSE
63 REPRESENTATIONAL RESPONSE BY SYMBOLIC SELECTION
64 REPRESENTATIONAL RESPONSE BY DUMMY CONTROL ACTIVATION
65 TASK-MANIPULATIVE RESPONSE, NON-DYNAMIC IN TIME AND FORCE
66 TASK-MANIPULATIVE RESPONSE, DYNAMIC AND INTERACTIVE
```

```
::: FEEDBACK PRESENTATION LOGIC
71 NOT APPLICABLE
72 SELECTS NEXT STIMULUS ITEM OR SEQUENCE
73 GIVES EVALUATION OF PRECEDING RESPONSE
74 SELECTS AND PRESENTS GUIDANCE INFO

::: RESPONSE EVALUATION LOGIC
81 NOT INTERNAL-DEPENDS ON INSTRUCTOR OR STUDENT EVALUATION
82 EVALUATION LIMITED TO STUDENT'S IMMEDIATE RESPONSE
83 EVALUATION EXTENDED TO A SET OF STUDENT RESPONSES
84 TOLERANCE LIMITS ON ACCEPTABLE STUDENT RESPONSE: FIXED
85 TOLERANCE LIMITS ON ACCEPTABLE STUDENT RESPONSE: VARIABLE


-- TASKS

::: ADMINISTRATIVE ROUTINE PAPERWORK
01 FORMS FILLING
02 DOCUMENT-FILE MANAGEMENT
03 DECODE-ENCODE
04 SCREEN-FILTER DISTRIBUTE
05 OTHER

::: ADMINISTRATIVE NON-ROUTINE PAPERWORK
11 CONSTRUCT MESSAGES-REPORTS
12 ANALYZE-INTERPRET
13 CONSTRUCT RECOMMENDATION-PROPOSAL
14 CONSTRUCT PLAN
15 OTHER

::: ADMINISTRATIVE OFFICE EQUIPMENT OPERATION
21 TYPEWRITER
22 TELEPHONE,ETC.
23 REPRODUCER
24 COMPUTER TERMINAL
25 TELETYPE
26 OTHER

::: INTERPERSONAL: INFORM-INSTRUCT-MANAGE
31 BRIEF-DEBRIEF
32 INSTRUCT-TRAIN
33 ASSIGN, MONITOR, COORDINATE
34 EVALUATE
35 OTHER

::: TECHNICAL PROCEDURES
41 SEQUENTIAL
42 STRATEGIC, ADAPTIVE
43 INTERPERSONAL,TEAM MEMBER
```

```
44 OTHER

*** TECHNICAL TYPE OF PROCEDURE(MAIN EMPHASIS)
51 SCAN-DETECT: SYMBOLIC (INCL. MAPS,RADAR,TRANSDUCED SIGNALS)
52 SCAN-DETECT: NATURAL
53 IDENTIFY: SYMBOLIC(INCL. TRANSDUCED SIGNALS)
54 IDENTIFY: NATURAL
55 INTERPRET:SYMBOLIC
56 INTERPRET: NATURAL
57 PERCEPTUAL-MOTOR
58 COGNITIVE OPERATIONS
59 MANUAL
60 COMMUNICATE

*** TECHNICAL WITH OR WITHOUT EQUIPMENT
61 WITH EQUIPMENT (PAPER IS EQUIPMENT)
62 WITHOUT EQUIPMENT

*** TECHNICAL DECIDE
71 DIAGNOSE-ANALYZE
72 SELECT-CHOOSE
73 UNDER STRESS OR LOAD
74 OTHER

*** TECHNICAL CONSTRUCT-REPAIR, PLAN
81 MANUAL CONSTRUCT OR REPAIR
82 COGNITIVE CONSTRUCT, PLAN

*** TECHNICAL TRACK-AIM-STEER
91 APPLICABLE
92 TIME STRESS
93 INFORMATION-LOAD STRESS

*** TOTAL DISCRIPTOR INDEX CARDS READ :    181 ***

MASTER DESCRIPTOR INDEX PROCESSING IS NOW TERMINATING
```

PROGRAM NAME : P17

ENTRY POINT : IHECMS (Execution Parameter Supplied by Executive RUNP17)

RUN EXECUTIVES(S) : RUNP17, NONAME, IFNAME, DEVICE, JCL2

USER OUTPUT VIA : Terminal, Offline Printer, or Both

USER PROMPTING : At Executive Level : Output Device Select

FUNCTION : Load One/More Project Files With Input Data Supplied on Cards

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
SYSPRINT	OUTPUT	N/A : THIS IS THE OFFLINE PRINTER			FA, 121, 121
CARDS	INPUT	25	DASD	Projectname/LOAD	F, 80, 80
REC	OUTPUT	27	DASD	Projectname/REC	F, 80, 80
REJ	OUTPUT	29	DASD	Projectname/REJ	F, 80, 80
REV	OUTPUT	28	DASD	Projectname/REV	F, 80, 80
ID	OUTPUT	26	DASD	Projectname/ID	F, 80, 80

The following files are not a functional component of the Range-Of-Effects (ROE) search process. Since they are card-image format, they can be loaded directly from Program P17 and printed from Program P1.

EXD	OUTPUT	33	DASD	Projectname/EXD	F, 80, 80
BPT	OUTPUT	34	DASD	Projectname/BPT	F, 80, 80
RKP	OUTPUT	35	DASD	Projectname/RKP	F, 80, 80
RRPJ	OUTPUT	36	DASD	Projectname/RRPJ	F, 80, 80
RRPK	OUTPUT	37	DASD	Projectname/RRPK	F, 80, 80
VQAL	OUTPUT	38	DASD	Projectname/VQAL	F, 80, 80
VARF	OUTPUT	39	DASD	Projectname/VARF	F, 80, 80
SCEN	OUTPUT	40	DASD	Projectname/SCEN	F, 80, 80
ALTP	OUTPUT	41	DASD	Projectname/ALTP	F, 80, 80

Discussion

This Program serves to generate those files of the ETAM system called "Project Files". In the current implementation, only those of card-image format are of concern. The following four card-image files are functional in the Range-of-Effects search portion of the ETAM system implementation:

Filetype	Usage
ID	Project Description
REC	Course Range-of-Effects (ROE) Search Descriptors
REV	Vehicle ROE Search Descriptors
REJ	Task ROE Search Descriptors

In addition to the above, if the control-card input so signifies, the following card-image Project Files may be initialized:

EXD	Search Extract Defaults
BPT	Benefit Pattern
RKP	Risk Profile
RRPJ	Risk Reduction Projects
RRPK	Risk Reduction Packages
VQAL	Variables Qualitative
VARF	Variable References
SCEN	Scenarios
ALTP	Alternate Projects

All of the above card-image files can be printed utilizing Program P1, either individually, or in a group using the "ALL" option.

Listing of EXECUTIVE = RUNP17

```
      &TYPE OFF
      &COMMENT THIS EXEC CONTROLS PROJECT FILE INIT/GENERATE
      &COMMENT SINGLE ENTRY PARM REQUIRED - PROJECT NAME
      &COMMENT EXEC = DEVICE WILL PROMPT FOR OUTPUT DEVICE
      &IF &INDEX EQ 1 &GOTO -STP1
 EXEC NONAME RUNP17
 -STP1 &ALPHA3 = &1
 EXEC IFNAME
      &SPACE
 EXEC DEVICE
      &SPACE
 EXEC JCL2
 FILEDEF SYSPRINT PTR RE FA LR 121 BL 121
 FILEDEF CARDS DSK &ALPHA3 LOAD RE F LR 80 BL 80
 LOAD P17 (CLEAR LIBE) PLILIB
 START IHECMS &ALPHA1 (BRIEF)
 FILEDEF * CLEAR
      &EXIT
```

Program P17 Specific User Instructions

Since this program processes only those files associated with an individual ETAM project in a single run, the name of the project must be provided by the User when invoking the RUNP17 executive. On the sample runs to follow, this is demonstrated using the ETAM project name "PN123".

The format (contents) of the input control file (File Number 25) dictates the loading process within Program P17. See the File Section description of File Number 25 for specific card formats. The following notes apply to the ordering of the control cards within the input file.

(1) Initialization of a particular file is signified in the control input by the occurance of a control card with the characters "**" in record positions one and two. The Project File filetype appears in record positions four through seven. All cards following this card up to the next "**" card or end-of-file will be loaded into the indicated filetype.

(2) More than one filetype may be initialized within one input control file.

(3) If a given file is to be initialized, and the file already exists, then the old copy is replaced with the new.

(4) If the first card of the control input is an "INIT" card, then any Project File not mentioned in the control input will be opened and one blank record inserted by Program P17.

Those files deemed "Project Files" in the ETAM system have been implemented in the NCSS time-sharing system as follows:

NCSS Filename	NCSS Filetype
The specific ETAM Project Name	ID, REC, REV, Etc.

Program P17 Error Messages

If the "INIT" option is used as in the instructions above, then the second card of the input control file must be an "**" type to signify the placement of the cards to follow. If the "INIT" option is omitted, then the first card must be an "**" type. If this condition is not met, the following error message will be output:

```
*** FIRST TITLE CARD IS MISSING ***
```

Once a "**" title card has been detected, the filetype name on the card will be verified; the following message test will appear if the filetype is illegal. This usually signifies a spelling error:

```
*** TITLE CARD FILENAME -filename- IS INVALID ***
```

The name in error is indicated in the message. All cards following this illegal title card will be printed on the selected output device(s), until the occurance of the next title card or end-of-file.

As files are initialized, Program P17 maintains a list of the files that have been processed within the current run. If an attempt is made to initialize the same filetype two or more times within the same run, the following error text will be output:

```
*** FILENAME -filename- IS DUPLICATED ***
```

All cards following this error will be printed up to the occurance of the next "**" title card or end-of-file.

Program P17 Sample Run Output

In the following example, the User has selected to print the contents of the initialization on the offline printer. The text shown below appears on the online terminal. Ten of the thirteen files have been provided card initialization; three were initialized with one blank card-image record. Since the blank initialization occured, the control input for this run contained an "INIT" card as the first card.

```
          RUNP17 PN123

  PROJECT NAMED PN123 DOES NOT YET EXIST,
  IS THIS CORRECT? (RESPOND YES OR QUIT)

  YES


  OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
  ENTER TERM, PRTR, BOTH, OR QUIT
  PRTR

  $EXECUTION:

  PROJECT FILE INITIALIZATION PROGRAM IS STARTING

  LOAD OF PROJECT FILETYPE : ID - PROJECT DESCRIPTION
  LOAD OF PROJECT FILETYPE : REC - COURSES ROE SEARCH ARGUMENTS
  LOAD OF PROJECT FILETYPE : BPT - BENEFIT PATTERN
  LOAD OF PROJECT FILETYPE : RKP - RISK PROFILE
  LOAD OF PROJECT FILETYPE : RRPJ - RISK REDUCTION PROJECTS
```

```
LOAD OF PROJECT FILETYPE : SCEN - SCENARIOS
LOAD OF PROJECT FILETYPE : RRPK - RISK REDUCTION PACKAGES
LOAD OF PROJECT FILETYPE : VQAL - VARIABLES QUALIFICATION
LOAD OF PROJECT FILETYPE : VARF - VARIABLE REFERENCES
LOAD OF PROJECT FILETYPE : ALTP - ALTERNATE PROJECTS

FILE : REJ  HAS BEEN INITIALIZED WITH ONE (1) BLANK RECORD

FILE : REV  HAS BEEN INITIALIZED WITH ONE (1) BLANK RECORD

FILE : EXD  HAS BEEN INITIALIZED WITH ONE (1) BLANK RECORD

*** ALL DATASETS INITIALIZED ***

*** TOTAL NUMBER INPUT CARDS READ :  461 ***

PROJECT FILE INITIALIZATION PROGRAM  IS NOW TERMINATING
```

PROGRAM NAME : P18

ENTRY POINT : P18

RUN EXECUTIVES(S) : STARTUP, START1, START2

USER OUTPUT VIA : None

USER PROMPTING : None

FUNCTION : Initializes (zeros) counts in Master Reference File (File Number 1)

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
MREF	OUTPUT	1	DASD	ETAM/MREF	F, 80, 80

Discussion

This file contains forty (40) binary halfword counts per record. Currently, there is only one record (one card-image) in use in this file. All forty quantities are zeroed by one run of P18, i.e., any previous contents will be destroyed.

Listing of EXECUTIVE = STARTUP

```
          P STARTUP EXEC

     &TYPE OFF
     &COMMENT THIS IS THE SYSTEM STARTUP EXEC
     &COMMENT NO INPUT PARAMETERS REQUIRED
-STEP1 STATE ETAM MREF P
     &IF &INDEX0 EQ 0 &GOTO -MDONE
LOAD P18 (CLEAR LIBE) PLILIB
FILEDEF MREF DSK ETAM MREF RE F LR 80 BL 80
START (BRIEF)
     &IF &INDEX0 EQ 0 &GOTO -MGO
-ER1 EXEC BROKE 1 &INDEX0
EXEC START1 4 MREF
     &GOTO -STEP2
-MGO EXEC START1 2 MREF
     &GOTO -STEP2
-MDONE EXEC START1 1 MREF
-STEP2 EXEC START2 CDESC SCDESC 2 1 8
-STEP3 EXEC START2 VDATA SVDATA 3 1 13
-STEP4 EXEC START2 VCDESC SVCDESC 4 1 13
-STEP5 EXEC START2 TDATA STDATA 5 1 13
-STEP6 EXEC START2 TDSCR STDSCR 6 1 13
     &EXIT
```

Listing of EXECUTIVE = START1

```
&TYPE OFF
&COMMENT P1 IS INTERNAL MESSAGE ROUTER, RANGES 1 TO 5
&COMMENT P2 IS FILETYPE NAME
&IF &1 EQ 5 &GOTO -NYET
&IF &1 EQ 4 &GOTO -BUST
&IF &1 EQ 3 &GOTO -RGO
&IF &1 EQ 2 &GOTO -GO
&PRINT FILE &2 HAS BEEN PREVIOUSLY INITILIZED
&EXIT
-GO &PRINT INITILIZE OF FILE &2 WAS SUCCESSFUL
&EXIT
-RGO &PRINT REINITILIZE OF FILE &2 WAS SUCCESSFUL
&EXIT
-BUST &PRINT FILE &2 INITILIZATION FAILED
&SPACE
&EXIT
-NYET &PRINT FILE &2 CANNOT BE INITILIZED (NO INPUT DATA)
&EXIT
```

Listing of EXECUTIVE = START2

```
          &TYPE OFF
     &COMMENT P1 IS FT OF OFFLINE READ INPUT FILE
     &COMMENT P2 IS FT OF NEW FILE TO BE CREATED
     &COMMENT P3 IS ERROR NUMBER, IF BREAKS
     &COMMENT P4 AND P5 ARE THE SORT PARMS
STATE ETAM &1 P
     &IF &INDEX0 NE 0 &GOTO -NONEW
ERASE .ETAM &1
     &STACK &4 &5
     &STACK KT
SORT ETAM &1 (BRIEF)
     &IF &INDEX0 EQ 0 &GOTO -SGO
-ERR EXEC BROKE &3 &INDEX0
EXEC START1 4 &2
     &EXIT &3
-SGO STATE ETAM &2 P
     &IF &INDEX0 NE 0 &GOTO -NOREG
     &INDEX1 = 3
ERASE ETAM &2 P
     &GOTO -SNEW
-NOREG &INDEX1 = 2
-SNEW ALTER .ETAM &1 P ETAM &2 P
EXEC START1 &INDEX1 &2
     &EXIT
-NONEW STATE ETAM &2 P
     &IF &INDEX0 EQ 0 &GOTO -STP1
EXEC START1 5 &2
     &EXIT
-STP1 EXEC START1 1 &2
     &EXIT
```

Program P18 Error Messages

None required.

Program P18 Sample Run Output

None generated by this program.

PROGRAM NAME : P20

ENTRY POINT : IHECMS (Execution Parameter Supplied by Executive RUNP20 at Entry

RUN EXECUTIVES(S) : RUNP20, DEVICE, JCL

USER OUTPUT VIA : Terminal Offline Printer, or Both

USER PROMPTING : At Executive Level: (1) Output Device Select
(2) Print Option Select (C, V, or T)

FUNCTION : Prints Course, Vehicle, or Task Entities and the Descriptors Associated With Each

Internal Filename (DDNAME)	Usage	Refer To File Reference Number	Media	External NCSS Filename/Filetype	Format
SYSPRINT	OUTPUT	N/A : THIS IS THE OFFLINE PRINTER			FA, 121, 121
MREF	INPUT	1	DASD	ETAM/MREF	F, 80, 80
DIND	DIRECT	3	DASD	ETAM/DIND	F, 80, 80
DESC	INPUT	6	DASD	ETAM/DESC	F, 213, 213
DCIN	INPUT	8	DASD	ETAM/DCIN	F, 760, 760
VDESC	INPUT	12	DASD	ETAM/VDESC	F, 213, 213
DVEH	INPUT	16	DASD	ETAM/DVEH	F, 750, 750
TDESC	INPUT	19	DASD	ETAM/TDESC	F, 213, 213
DTASK	INPUT	23	DASD	ETAM/DTASK	F, 1500, 1500
VEHS	DIRECT	15	DASD	ETAM/VEHS	F, 80, 80
TASKS	DIRECT	22	DASD	ETAM/TASKS	F, 80, 80

Discussion

This program provides print facilities for the three types of ETAM entities - Courses, Vehicles, and Tasks. The entity type and output device type are selected by the User at the executive level.

Printer output from Program P20 is of the general format: entity description followed by all descriptors assigned to that particular entity. The order of output for Courses is by CIN number; for Vehicles, Stock Number order is maintained. For Tasks, the output is ordered by the contents of the 13-byte field of concatenated Rate, Rank, and Jobtask Number.

Listing of EXECUTIVE = RUNP20

```
      &TYPE OFF
      &COMMENT G.P. PRINT EXEC FOR COURSES, VEHICLES, TASKS ;
      &COMMENT PRINTS DESCRIPTORS ASSIGNED TO EACH ENTITY
      &COMMENT ENTRY/DEVICE PROMPTS DONE HERE AND IN EXEC = DEVICE
      &SPACE
      &PRINT DESCRIPTOR PRINT PROGRAM IS STARTING
      &SPACE
      &PRINT ENTER SINGLE CHARACTER OPTION (OR QUIT TO TERMINATE)
      &PRINT C = PRINT ALL DESCRIPTORS FOR COURSES, SORTED BY CIN
      &PRINT V = PRINT ALL DESCRIPTORS ASSIGNED TO VEHICLES
      &PRINT T = LIST DESCRIPTORS ASSIGNED TO TASK DATA
-STP1 &READ ARGS
      &IF &1 EQ QUIT &QUIT
      &ALPHA0 = &1
      &IF &1 EQ C &GOTO -STP2
      &IF &1 EQ V &GOTO -STP2
      &IF &1 EQ T &GOTO -STP2
      &SPACE
      &PRINT UNABLE TO DETERMINE ENTRY TYPE - TRY AGAIN
      &SPACE
      &GOTO -STP1
-STP2 EXEC DEVICE
      &ALPHA2 = &ALPHA0 !! &ALPHA1
-STP3 &SPACE
EXEC JCL
FILEDEF SYSPRINT PTR RE FA LR 121 BL 121
LOAD P20 (CLEAR LIBE) PLILIB
START IHECMS &ALPHA2 (BRIEF)
FILEDEF * CLEAR
      &EXIT
```

Program P20 Error Messages

Program P20 can produce the following error text (two lines):

```
** ERROR NUMBER   XX HAS OCCURED **
```

(Followed by a descriptive content line - varies with message number, as indicated below)

The error numbers associated with the above message have the following significance:

01 - The entry parameter supplied to Program P20 at entry time was not one of the three characters "C", "V", or "T".

The 2nd error line will contain the actual character(s) provided as an entry parameter by the external executive.

02 - A pointer discrepancy has been detected in the Descriptor Master Index file (File Number 3).

The second error line containes the expected and actual pointer values of concern.

03 - Pointer to the ETAM/DIND Descriptor Master Index file is outside the bounds of the file.

Second error line = same as that for error number 02.

04 - Number of categories in the ETAM/DIND file is greater than 15.

Second error line, same as for error number 02.

Program P20 Specific User Instructions

Three print options exist for the User in Program P20. These options are exercised by the selection of a single character to be entered via online terminal. The options are as follows:

Character	Meaning
"C"	Select Courses and their related descriptors for output
"V"	Select Vehicles, etc.
"T"	Select Tasks.

Program P20 Sample Run Output - Course Selection

Selection of the "C" options results in output as presented in the following sample. Shown are the prompting sequences and a few sample lines to illustrate the general format for Course output.

```
          RUNP20

DESCRIPTOR PRINT PROGRAM IS STARTING

ENTER SINGLE CHARACTER OPTION (OR QUIT TO TERMINATE)
C = PRINT ALL DESCRIPTORS FOR COURSES, SORTED BY CIN
V = PRINT ALL DESCRIPTORS ASSIGNED TO VEHICLES
T = LIST DESCRIPTORS ASSIGNED TO TASK DATA
C

OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
ENTER TERM, PRTR, BOTH, OR QUIT
TERM

$$EXECUTION:

PAGE     1 : COURSE  DESCRIPTOR ASSIGNMENTS

CIN : A 000111

        ::: REFERENCE KNOWLEDGE
               01 SYSTEM PURPOSES
               03 CONTEXTS OF OPERATION
               04 ORGANIZATIONAL RULES

        ::: ENABLING KNOWLEDGE
               11 NOMENCLATURE, IDENTIFICATION, LOCATION
               12 PROCEDURAL DESCRIPTIONS
               13 JOB RELEVANT FACTS,RULES

        ::: TASK FORMATS
               20 PROCEDURE FORMATS
               21 DECISION FORMATS
               22 CONSTRUCTION FORMATS

        ::: GROSS JOB CATEGORIES
               33 COMMAND

        ::: OBJECTIVE TASK VARIABLES AS MANIFEST IN THE TRAINING
               42 EQUIPMENT & OBJECTS USED: SYMBOLIC
               45 ENVIRONMENTS IN WHICH TASK IS TRAINED: SYMBOLIC
               48 TOOLS/INSTRUMENTS USED IN TRAINING: SYMBOLIC
               53 CRITERIA OF TASK PERFORMANCE: SYMBOLIC
          *** TASK FUNCTIONS DOMINANT IN TRAINING
                  65 DECIDE
                  66 CONSTRUCT, PLAN

          *** STAGE OF LEARNING
                  82 TASK FORMATS AT CONCEPTUAL LEVEL

   CIN : A 2E0013

          *** REFERENCE KNOWLEDGE
                  01 SYSTEM PURPOSES
                  02 ORGANIZATIONAL ROLES
                  03 CONTEXTS OF OPERATION
```

... Output interrupted at this point ...

Program P20 Sample Run Output - Vehicles Selection

Following is an illustration of the "V" option output:

```
          RUNP20

   DESCRIPTOR PRINT PROGRAM IS STARTING

   ENTER SINGLE CHARACTER OPTION (OR QUIT TO TERMINATE)
   C = PRINT ALL DESCRIPTORS FOR COURSES, SORTED BY CIN
   V = PRINT ALL DESCRIPTORS ASSIGNED TO VEHICLES
   T = LIST DESCRIPTORS ASSIGNED TO TASK DATA
   V

   OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
   ENTER TERM, PRTR, BOTH, OR QUIT
   TERM

   SSEXECUTION:

   PAGE     1 : VEHICLE DESCRIPTOR ASSIGNMENTS

   6910LLC00046 / 12CD-17    / MATH, VECTOR ACCELERATION DEMONSTRATOR T/A

          *** VEHICLE TYPES
                 03 ANIMATED GRAPHICS

          *** CLASS OF TRAINING OBJECTIVE
                 11 REFERENCE KNOWLEDGE

          *** VEHICLE PROPERTIES
                 21 VISUAL
```

... Output terminated at this point ...

Program P20 Sample Run Output - Tasks Option

Selection of the "T" option results inoutput that is in the following format:

```
          RUNP20

 DESCRIPTOR PRINT PROGRAM IS STARTING

 ENTER SINGLE CHARACTER OPTION (OR QUIT TO TERMINATE)
 C = PRINT ALL DESCRIPTORS FOR COURSES, SORTED BY CIN
 [illegible]= PRINT ALL DESCRIPTORS ASSIGNED TO VEHICLES
 T = LIST DESCRIPTORS ASSIGNED TO TASK DATA
 T

 OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
 ENTER TERM, PRTR, BOTH, OR QUIT
 TERM

 $EXECUTION:

 PAGE     1 : JOBTASK DESCRIPTOR ASSIGNMENTS

 AN     /     / 20350     AIRCRAFT UNDER EMERGENCY CONDITIONS

      ::: TECHNICAL PROCEDURES
          41 SEQUENTIAL
          42 STRATEGIC, ADAPTIVE

      ::: TECHNICAL TYPE OF PROCEDURE(MAIN EMPHASIS)
          57 PERCEPTUAL-MOTOR

      ::: TECHNICAL WITH OR WITHOUT EQUIPMENT
          61 WITH EQUIPMENT (PAPER IS EQUIPMENT)

      ::: TECHNICAL DECIDE
          73 UNDER STRESS OR LOAD


 AN     /     / 20351     IDENTIFY MARKINGS INDICATING DANGEROUS AREAS

      ::: TECHNICAL TYPE OF PROCEDURE(MAIN EMPHASIS)
          53 IDENTIFY: SYMBOLIC(INCL. TRANSDUCED SIGNALS)


 AN     /     / 20352     IDENTIFY TOXIC PROPERTIES OF CLEANING MATERIALS

      ::: TECHNICAL TYPE OF PROCEDURE(MAIN EMPHASIS)
          54 IDENTIFY: NATURAL


 AN     /     / 20368     RECOGNIZE ARMED EJECTION SEATS

      ::: TECHNICAL TYPE OF PROCEDURE(MAIN EMPHASIS)
          54 IDENTIFY: NATURAL
```

... Output terminated at this point ...

SECTION D.3

NCSS EXECUTIVE SEQUENCES ASSOCIATED WITH MULTIPLE PROGRAMS

Each program of the ETAM system has an associated NCSS executive (exec) routine. These executives are listed in the Section covering the program of concern.

The executives within this Section are, in general, multiple usage modules not associated with a single ETAM program; a brief description and a listing of each follows.

EXECUTIVE = BROKE

This is a general-purpose routine used to print a standardized error message. The routine is designed to be invoked by a higher-level executive. Return is always to the caller-executive routine.

```
&TYPE OFF
&COMMENT P1 IS ERROR NUMBER
&COMMENT P2 IS INDEX0 VALUE THAT CAUSED ERROR
&SPACE
&PRINT *** ERROR NUMBER &1 HAS OCCURED ; INDEX0 VALUE WAS &2 ***
&SPACE
&EXIT
TOP
```

EXECUTIVE = DEVICE

Many of the ETAM programs have been designed to direct output to the User's online terminal, offline printer, or both. This executive function is to prompt the User in the selection of the desired device(s). If the User elects the QUIT option, forced termination of the executive sequence at all levels follows.

```
      &TYPE OFF
      &COMMENT PROMPTING EXEC FOR DEVICE
      &COMMENT NO ENTRY ARGUMENTS REQUIRED
      &COMMENT IF QUIT ENTERED, EXEC SEQUENCE TERMINATED
      &COMMENT NORMAL RETURN WITH &ALPHA1 SET TO ENTRY
-STP1 &SPACE
      &PRINT OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, BOTH, OR QUIT?
      &PRINT ENTER TERM, PRTR, BOTH, OR QUIT
```

```
-STP2 &READ ARGS
      &IF &1 EQ QUIT &QUIT
      &ALPHA1 = &1
      &IF &1 EQ BOTH &EXIT
      &IF &1 EQ PRTR &EXIT
      &IF &1 EQ TERM &EXIT
      &SPACE
      &PRINT UNABLE TO DETERMINE ENTRY TYPE - TRY AGAIN
      &SPACE
      &GOTO -STP2
```

EXECUTIVE = IFNAME

Those programs associated with a specific ETAM project must be furnished the name of the project. No operation can ensue, if the project name is omitted. This executive determines the existence of the project name provided.

```
        P IFNAME EXEC

     &TYPE OFF
     &COMMENT EXEC TO VALIDATE THE EXISTENCE OF A USER-
     &COMMENT SUPPLIED PROJECT NAME
     &COMMENT ENTRY REQUIRES GLOBAL ALPHA VARIABLE NO. 3
     &COMMENT TO BE INITILIZED TO THE PROJECT FILE NAME
     &SPACE
STATE &ALPHA3 ID P
     &IF &INDEX0 EQ 0 &GOTO -YES
-NO &PRINT PROJECT NAMED &ALPHA3 DOES NOT YET EXIST,
     &GOTO -STP1
-YES &PRINT PROJECT NAMED &ALPHA3 ALREADY EXISTS,
-STP1 &PRINT IS THIS CORRECT?  (RESPOND YES OR QUIT)
     &SPACE
     &READ ARGS
     &IF &1 EQ YES &EXIT
     &IF &1 EQ QUIT &QUIT
     &SPACE
     &PRINT CANNOT DETERMINE LAST ENTRY - TRY AGAIN
     &SPACE
     &GOTO -STP1
```

EXECUTIVE = JCL

This executive provides the FILEDEF statements required by an ETAM program making reference to any of the abbreviated data bases.

```
        &TYPE OFF
        &COMMENT G.P. FILEDEF EXEC FOR ETAM SYSTEM FILES
        &COMMENT NO ENTRY/EXIT PARAMETERS OR ARGUMENTS
   FILEDEF MREF DSK ETAM MREF RE F LR 80 BL 80
   FILEDEF DIND DSK ETAM DIND RE F LR 80 BL 80
   FILEDEF COURSE DSK ETAM COURSE RE F LR 80 BL 80
   FILEDEF DCIN DSK ETAM DCIN RE F LR 760 BL 760
   FILEDEF DCDP DSK ETAM DCDP RE F LR 1200 BL 1200
   FILEDEF DESC DSK ETAM DESC RE F LR 213 BL 213
   FILEDEF VEHS DSK ETAM VEHS RE F LR 80 BL 80
   FILEDEF DVEH DSK ETAM DVEH RE F LR 750 BL 750
   FILEDEF VDESC DSK ETAM VDESC RE F LR 213 BL 213
   FILEDEF TASKS DSK ETAM TASKS RE F LR 80 BL 80
   FILEDEF DTASK DSK ETAM DTASK RE F LR 1500 BL 1500
   FILEDEF TDESC DSK ETAM TDESC RE F LR 213 BL 213
        &EXIT
```

EXECUTIVE = JCL2

Similar to the JCL executive, this executive provides FILEDEF statements for programs requiring access to the files of a specific ETAM project.

```
        &TYPE OFF
        &COMMENT G.P. FILEDEFS FOR PROJECT FILES
        &COMMENT SINGLE ENTRY PARM IS PROJECT FILETYPE IN
        &COMMENT GLOBAL ALPHA VARIABLE 3
   FILEDEF ID DSK &ALPHA3 ID RE F LR 80 BL 80
   FILEDEF REC DSK &ALPHA3 REC RE F LR 80 BL 80
   FILEDEF REJ DSK &ALPHA3 REJ RE F LR 80 BL 80
   FILEDEF REV DSK &ALPHA3 REV RE F LR 80 BL 80
   FILEDEF EXD DSK &ALPHA3 EXD RE F LR 80 BL 80
   FILEDEF BPT DSK &ALPHA3 BPT RE F LR 80 BL 80
   FILEDEF RKP DSK &ALPHA3 RKP RE F LR 80 BL 80
   FILEDEF RRPJ DSK &ALPHA3 RRPJ RE F LR 80 BL 80
   FILEDEF RRPK DSK &ALPHA3 RRPK RE F LR 80 BL 80
   FILEDEF VOAL DSK &ALPHA3 VOAL RE F LR 80 BL 80
   FILEDEF VARF DSK &ALPHA3 VARF RE F LR 80 BL 80
   FILEDEF SCEN DSK &ALPHA3 SCEN RE F LR 80 BL 80
   FILEDEF ALTP DSK &ALPHA3 ALTP RE F LR 80 BL 80
   FILEDEF RESU DSK &ALPHA3 RESU RE FB LR 16 BL 800
   FILEDEF REE DSK &ALPHA3 REE RE FB LR 16 BL 800
   FILEDEF SARG DSK &ALPHA3 SARG RE F LR 240 BL 240
        &EXIT
```

EXECUTIVE = NONAME

If a program associated with an ETAM project file is invoked without supplying the name of the project, this executive will diagnose the error. A message will be output to inform the User, and action at all levels will be terminated.

```
&TYPE OFF
&COMMENT THIS EXEC FOR ERROR MESSAGE AND FORCED TERMINATE
&COMMENT IN THE EVENT OF NO PROJECT NAME SUPPLIED.
&COMMENT SINGLE ENTRY PARM IS NAME OF EXEC ENDING.
&SPACE
&PRINT *** NO PROJECT NAME SUPPLIED BY USER,
&PRINT     EXEC &1 IS TERMINATING ABNORMALLY ***
&SPACE
&QUIT
```

EXECUTIVE = ETAM

This is the ETAM master control executive as discussed in Section III. Figure III-18 depicts the control logic implemented into this executive. The text output to the online terminal is illustrated in Figure III-19. Following this executive listing is a listing of NCSS file ETAM/TEXT1 - this contains output text to be used by the ETAM executive.

```
    &TYPE OFF
    &COMMENT THIS IS THE OVERALL SYSTEM EXEC
    &COMMENT NO ENTRY PARMS REQUIRED
-S1 &SPACE
    &PRINT ENTER PROJECT IDENTIFIER, E.G., PN123 (5 CHARACTERS)
    &SPACE
    &READ ARGS
    &SPACE
    &IF &INDEX EQ 1 &GOTO -S2
    &PRINT CANNOT DECODE LAST ENTRY - TRY AGAIN
    &GOTO -S1
-[illegible] &IF &1 EQ QUIT &QUIT
    &ALPHA5 = &1
STATE &ALPHA5 ID P
    &IF &INDEX0 EQ 0 &GOTO -S5
-S3 &SPACE
    &PRINT PROJECT FILE DOES NOT EXIST
    &PRINT GENERATE NEW FILE? ? ?
    &PRINT 1) YES
    &PRINT 2) NO
    &PRINT ENTER: 1 OR 2
    &SPACE
    &READ ARGS
    &SPACE
    &IF &1 EQ 2 &GOTO -S1
    &IF &1 EQ QUIT &QUIT
    &IF &1 EQ 1 &GOTO -S4
    &PRINT CANNOT DECODE LAST ENTRY - TRY AGAIN
    &GOTO -S3
-S4 EXEC RUNP17 &ALPHA5
-S5 &SPACE
    &PRINT SELECT OPTION.
    &PRINT 1) ROE
    &PRINT 2) ASSESS
    &PRINT 3) PRINT
    &PRINT 4) NEW PROJECT
    &PRINT 5) INSTRUCTIONS
    &PRINT 6) QUIT
    &PRINT ENTER: 1 ... 6
    &SPACE
    &READ ARGS
    &SPACE
    &IF &1 EQ QUIT &QUIT
    &IF &1 EQ 6 &QUIT
    &IF &1 EQ 4 &GOTO -S1
    &IF &1 EQ 5 &GOTO -MENU
    &IF &1 EQ 3 &GOTO -PGM1
    &IF &1 EQ 2 &GOTO -ASSESS
    &IF &1 EQ 1 &GOTO -ROE
    &PRINT CANNOT DECODE LAST ENTRY - TRY AGAIN
    &GOTO -S5
-PGM1 EXEC RUNP1 &ALPHA5
    &GOTO -S5
-MENU PRINTF ETAM TEXT1
    &GOTO -S5
-ROE &SPACE
    &PRINT RANGE-OF-EFFECT, SELECT OPTION
    &PRINT 1) EDIT DESCRIPTOR
    &PRINT 2) SEARCH
    &PRINT 3) PRINT SEARCH
    &PRINT 4) EDIT SEARCH
    &PRINT 5) PRINT EXTRACT
    &PRINT 6) QUIT
    &PRINT ENTER: 1 ... 6
    &SPACE
    &READ ARGS
    &SPACE
    &IF &1 EQ QUIT &QUIT
    &IF &1 EQ 6 &QUIT
    &IF &1 EQ 1 &GOTO -PGM2
    &IF &1 EQ 2 &GOTO -PGM3
    &IF &1 EQ 3 &GOTO -PGM5A
    &IF &1 EQ 4 &GOTO -PGM5B
    &IF &1 EQ 5 &GOTO -PGM5C
    &PRINT CANNOT DECODE LAST ENTRY - TRY AGAIN
    &GOTO -ROE
```

```
-PGM2 EXEC RUNP2 &ALPHA5
    &GOTO -MROE
-PGM3 EXEC RUNP3 &ALPHA5
    &GOTO -MROE
-PGM5A EXEC RUNP5A &ALPHA5
    &GOTO -MROE
-PGM5B EXEC RUNP5B &ALPHA5
    &GOTO -MROE
-PGM5C EXEC RUNP5C &ALPHA5
-MROE &SPACE
    &PRINT CONTINUE ROE TO SELECT OTHER ROUTINES? ? ?
    &PRINT 1) YES
    &PRINT 2) NO
    &PRINT ENTER: 1 OR 2
    &SPACE
    &READ ARGS
    &SPACE
    &IF &1 EQ QUIT &QUIT
    &IF &1 EQ 1 &GOTO -ROE
    &IF &1 EQ 2 &GOTO -S5
    &PRINT CANNOT DECODE LAST ENTRY - TRY AGAIN
    &GOTO -MROE
-ASSESS &SPACE
    &PRINT ASSESSMENT - SELECT OPTION
    &PRINT 1) MODEL
    &PRINT 2) RISK
    &PRINT 3) DECISION
    &PRINT 4) FINANCIAL
    &PRINT 5) QUIT
    &PRINT ENTER: 1 ... 5
    &SPACE
    &READ ARGS
    &SPACE
    &IF &1 EQ QUIT &QUIT
    &IF &1 EQ 5 &QUIT
    &IF &1 EQ 1 &GOTO -NONE
    &IF &1 EQ 2 &GOTO -NONE
    &IF &1 EQ 3 &GOTO -NONE
    &IF &1 EQ 4 &GOTO -NONE
    &PRINT CANNOT DECODE LAST ENTRY -TRY AGAIN
    &GOTO -ASSESS
-NONE &SPACE
    &PRINT PROGRAM NOT INSTALLED.
-MASS &SPACE
    &PRINT CONTINUE ASSESS TO SELECT OTHER ROUTINES? ? ?
    &PRINT 1) YES
    &PRINT 2) NO
    &PRINT ENTER: 1 OR 2
    &SPACE
    &READ ARGS
    &SPACE
    &IF &1 EQ QUIT &QUIT
    &IF &1 EQ 1 &GOTO -ASSESS
    &IF &1 EQ 2 &GOTO -S5
    &PRINT CANNOT DECODE LAST ENTRY - TRY AGAIN
    &GOTO -MASS
```

FILE = ETAM/TEXT1

This file contains text to be output by the master control executive named ETAM.

```
              P ETAM TEXT1

   ETAM INSTRUCTIONS

   OPTIONS 1 THRU 6 PERMIT SELECTION OF THE FOLLOWING ETAM FUNCTIONS.

   1    WILL CAUSE ENTRY INTO THE ETAM RANGE-OF-EFFECT FUNCTION FOR:
             RUNNING DESCRIPTOR EDIT PROGRAM
             SEARCHING ABBREVIATED DATA BASES
             PRINTING SEARCH RESULTS
             RUNNING SEARCH EDIT PROGRAM
             RUNNING PRINT EXTRACT PROGRAM

   2    WILL CAUSE ENTRY INTO THE ETAM ASSESSMENT FUNCTION FOR:
             SETTING UP MODEL INPUTS FOR MODEL RUN
             PACKAGING RISK REDUCTION PROJECTS
             RUNNING DECISION ANALYSIS PROGRAMS
             RUNNING FINANCIAL ANALYSIS PROGRAMS

   3    WILL PERMIT PRINTING OF PROJECT DATA BASE FILES

   4    WILL CAUSE A RETURN TO THE BEGINNING OF THE ETAM CONTROL
        EXEC SO THAT A DIFFERENT PROJECT ID CAN BE ENTERED

   5    WILL CAUSE THESE ETAM INSTRUCTIONS TO BE REPRINTED

   6    WILL END ETAM EXEC CONTROL AND RETURN TO NCSS CONTROL
```

SECTION D.4

ETAM FILE FORMATS, AS IMPLEMENTED

The current ETAM system has been implemented to provide those features up to and including the Range-of-Effects search process. This ahs resulted in the identification of forty-one (41) unique data files.

In conformity with the rules of the NCSS time-sharing vendor's online system, a file is made unique by establishing a unique "filename" and/or "filetype". Two filenames are used by the ETAM system, as follows:

Filename = ETAM:	For those applications and data required by all ETAM projects in common. For instance, all ETAM projects access the same Abbreviated Course data base.
Filename = Project name:	These files are unique to each given ETAM project. Convention dictates that this project name be of the format "PNxxx" where "xxx" is numeric.

FILE REFERENCE NUMBER : 1

NCSS FILENAME/FILETYPE : ETAM/MREF

MEDIA : DASD

FORMAT : F, 80, 80

CREATED/GENERATED BY : Program P18 (zeros contents); for value insertion, see table, below

CONTENTS PRINTED BY : Not Applicable

FILE FUNCTION : Record counts are saved in this file to be used, as required, by various programs of the ETAM system.

Record Position	Data Name	Length	Type	Value/Comments
1-2	Fixed Binary Count	2	Fixed Bin	See below, for meaning
3-80	Repeat of (1-2) Above, Occurs 40 Times Total	78	Fixed Bin	

This file contains counts to be used as required by the various programs of the ETAM system. MREF contains a single 80-byte (card image) record; as indicated above, space has been reserved for forty (40) unique fixed binary counts. Current usage is indicated in the table below. Indicated are the programs in which the individual quantities are established and the quantity meaning. Note that some are 0-origin (zero-origin), whereas others are 1-origin (one-origin) in their context.

Subscript	Count Origin	Count Related to Filetype	File Ref	Set in Program	Count Meaning
1	0	COURSE	7	P7	Total number of records in the Abbreviated course file
2	0	DCIN	8	P7	Maximum subscript used in the course file CIN directory
3	0	DCDP	9	P7	Total number of entries in the course file CDP directory; this quantity should correspond to the subscript (1) value, above
4	0	DIND	3	P16	Highest subscript used in the Descriptor Master Inde

Subscript	Count Origin	Count Related to Filetype	File Ref	Set in Program	Count Meaning
5	1	SCDESC	5	P7A	Total number of course file descriptor cards
6	1	DESC	6	P7A	Total number course descriptor records
7	0	VEHS	15	P8	Total number records in the Abbreviated vehicle data file
8	0	DVEH	16	P8	Maximum subscript set in the vehicle file directory
9	1	VDESC	12	P8	Total number of records in the vehicle descriptor file
10	1	SVCDESC	11	P8	Total number vehicle descriptor cards
11	0	TASKS	22	P9	Total number record in the Abbreviated vehicle file
12	0	DTASK	23	P9	Maximum subscript used in the task file directory
13	1	TDESC	19	P9	Total number records in the task descriptor file
14	1	STDSCR	18	P9	Total number task descriptor cards

FILE REFERENCE NUMBER : 2
NCSS FILENAME/FILETYPE : ETAM/DICRDS
MEDIA : DASD
FORMAT : F, 80, 80
CREATED/GENERATED BY : Not Applicable
CONTENTS PRINTED BY : Not Applicable
FILE FUNCTION : Input card deck for the Descriptor Master Index Build Program P15

Record Position	Data Name	Length	Type	Value/Comments
1-2	Field A	2	Char	
3	Filler	1	Char	Blank
4-80	Field B	77	Char	

One input card deck contains descriptor decks for all three entities, in following order: Courses, Vehicles, Tasks.

One descriptor deck for a single entity is composed as follows:

Card #1 Field A = '**'
Field B = 'COURSES', 'VEHICLE', or 'TASKS'; left justified; remainder of Field B is blank

Card #2 Field A = Descriptor Number
Field B = Descriptor Title

The combination of category card (#2, above) followed by as many descriptor cards (#3, above) as appropriate is repeated.

All descriptor numbers must be in ascending sort order, starting with number = one.

FILE REFERENCE NUMBER : 3
NCSS FILENAME/FILETYPE : ETAM/DIND
MEDIA : DASD
FORMAT : F, 80, 80
CREATED/GENERATED BY : Program P16
CONTENTS PRINTED BY : Program P16, P7B, P8A, P9A, P20
FILE FUNCTION : Descriptor Master Index

The following is a map of the Master Descriptor Index in memory. File number 3 is card image, theref[illegible] 4 80-byte records are required to load the following structure.

Records Nos. 0 thru 3 are the Course index; records 4 thru 7 load the Vehicle index; records 8 thru 11 load the Task index.

The Master Index description cards (see File number 2) are then loaded into this file sequentially, starting at record location 12.

Record Position	Data Name	Length	Type	Value/Comments
1-2	Start Card Number for this Entity Category	2	Fixed Bin	Card number = relative number within this file
3-4	End Card Number for this Entity	2	Fixed Bin	
5-6	Total Number of Descriptor Categories for this Entity	2	Fixed Bin	
7-8	Total Number of Descriptors Active for this Entity	2	Fixed Bin	
9-10	Pointer to 1st Category Card for this Entity	2	Fixed Bin	
11-12	Start Descriptor Number within the Current Category	2	Fixed Bin	
13-14	Last Descriptor Number within Current Category	2	Fixed Bin	
15-98	Pos (9-14) Repeated, Occurs 15 Times Total	84	Fixed Bin	Unused portion of this array is zeroed

FILE REFERENCE NUMBER : 4, 5

NCSS FILENAME/FILETYPE : (4) ETAM/CDESC; (5) ETAM/SCDESC

MEDIA : DASD

FORMAT : F, 80, 80

CREATED/GENERATED BY : Not Applicable

CONTENTS PRINTED BY : Not Applicable

FILE FUNCTION : Input card deck with Descriptors as assigned to course data

The CDESC card file is the data as transmitted to the NCSS time-sharing vendor. This file may be in unsorted format.

The SCDESC file has been sorted into ascending sequence by CIN number. This sort is accomplished in the executive sequence STARTUP.

Record Position	Data Name	Length	Type	Value/Comments
1-8	CIN Number	8	Char	
9	Filler	1	Char	Blank
10-11	2-Digit Descriptor	2	F(2)	P'99'
12-80	Pos 9-11, Repeated 23 times			

Notes - Max of 24 descriptors per card; if full card not required, leave remainder blank.

- Each descriptor is a two-digit number; supply lead zero, if applicable.

FILE REFERENCE NUMBER : 6

NCSS FILENAME/FILETYPE : ETAM/DESC

MEDIA : DASD

FORMAT : F, 213, 213

CREATED/GENERATED BY : Program P7A

CONTENTS PRINTED BY : Program P20 (Option C)

FILE FUNCTION : Descriptor data to match the contents of the Abbreviated course data file (File Number 7)

Record Position	Data Name	Length	Type	Value/Comments
1-8	CIN Number	1	Char	
9-13	Filler	5	Char	Blank
14-213	Descriptor Flags, Occurs (100) Times	200	Fixed Bin	Initially, all flags are zeroed

Note - Descriptor Flags (subscripted 1 to 100, incl.) are NON-ZERO for those descriptors applicable to the Course keyed by the CIN portion of the record.

FILE REFERENCE NUMBER : 7
NCSS FILENAME/FILETYPE : ETAM/COURSE
MEDIA : DASD
FORMAT : F, 80, 80
CREATED/GENERATED BY : Program P14
CONTENTS PRINTED BY : Program P7B
FILE FUNCTION : Abbreviated Course Data File

Record Position	Data Name	Length	Type	Value/Comments
1-4	CDP	4	Char	Blank, if missing
5-12	CIN	8	Char	
13-28	Course Short Title	16	Char	
29-32	NOBC	4	Char	
33-36	NEC	4	Char	
37-38	Priority Code	2	Char	
39-41	RMS Cost Code	3	Char	Last 3 digits only
42-43	Type Course	2	Char	
44	SVC Code	1	Char	
45	Method of Instruction	1	Char	
46	Status Code	1	Char	
47-51	Status Date	5	P	P'(5)9'
52	TRAPS Indicator	1	Char	
53-57	TPC	5	Char	
58-60	Established Attrition Rate	3	P	P'99V9'
61-63	Established Setback Rate	3	P	P'99V9'
64-66	Course Length	3	P	P'999'
67-69	Total Hours	3	P	P'999'

Record Position	Data Name	Length	Type	Value/Comments
70-72	Lab Hours	3	P	P'999'
73-79	Cost Per AOB*	7	P	P'(5)9V99'
80	Filler	1	Char	Blank

*This field currently not loaded; not available on the NITRAS MCRF Extract Tape.

FILE REFERENCE NUMBER : 8

NCSS FILENAME/FILETYPE : ETAM/DCIN

MEDIA : DASD

FORMAT : F, 760, 760

CREATED/GENERATED BY : Program P7

CONTENTS PRINTED BY : Not Applicable

FILE FUNCTION : Abbreviated Course File Directory, ordered by CIN Number

Record Position	Data Name	Length	Type	Value/Comments
1-8	CIN	8	Char	
9-10	Pointer to Relative Record in Abbreviated Course File (File Number 7)	2	Fixed Bin	
11-760	Repeat of Pos 1-10, Occurs (76) Times Total	750		Unused elements of the Array are blanked (CIN) or zeroed (pointer)

FILE REFERENCE NUMBER : 9
NCSS FILENAME/FILETYPE : ETAM/DCDP
MEDIA : DASD
FORMAT : F, 1200, 1200
CREATED/GENERATED BY : Program P7
CONTENTS PRINTED BY : Not Applicable
FILE FUNCTION : Abbreviated Course File Directory, ordered by CDP Number

Record Position	Data Name	Length	Type	Value/Comments
1-4	CDP	4	Char	
5-6	Pointer to Related Record in Abbreviated Course File (File Number 7)	2	Fixed Bin	
7-1200	Repeat of Pos (1-6), Occurs 200 Times Total	1194		Unused elements in the array are blanked (CDP) and zeroed (pointer)

FILE REFERENCE NUMBER : 10, 11

NCSS FILENAME/FILETYPE : (10) ETAM/VCDESC; (11) ETAM/SVCDESC

MEDIA : DASD

FORMAT : F, 80, 80

CREATED/GENERATED BY : Not Applicable

CONTENTS PRINTED BY : Not Applicable

FILE FUNCTION : Input card deck with Descriptors as assigned to Vehicle data

The VCDESC card file is the Vehicle Descriptor data as transmitted to the NCSS time-sharing vendor. This file may be in unsorted format.

The SVCDESC file has been sorted into ascending sequence by Vehicle Stock Number. The sort is accomplished by running the executive sequence STARTUP.

Record Position	Data Name	Length	Type	Value/Comments
1-13	Vehicle Stock Number	13	Char	
14	Filler	1	Char	Blank
15-16	2-Digit Descriptor	2	F(2)	p'99'
17-79	Pos 14-16, Repeated 21 Times	63		
80	Filler	1	Char	Blank

Notes - Max of 22 descriptors per card; if full card not required, leave remainder blank

- Each descriptor is a two-digit number; suppy lead zero, if applicable

FILE REFERENCE NUMBER : 12
NCSS FILENAME/FILETYPE : ETAM/VDESC
MEDIA : DASD
FORMAT : F, 213, 213
CREATED/GENERATED BY : Program P8
CONTENTS PRINTED BY : Program P20 (Option V)
FILE FUNCTION : Descriptor data to match the contents of the Abbreviated Vehicle data file (File Number 15)

Record Position	Data Name	Length	Type	Value/Comments
1-13	Vehicle Stock Number	13	Char	
14-213	Descriptor Flags, Occurs (100) Times	200	Fixed Bin	Initially, all 100 flags are zeroed

Note - Descriptor Flags (subscripted 1 to 100, incl.) are NON-ZERO for those descriptors applicable to the Vehicle keyed by the first 13 characters of the record.

FILE REFERENCE NUMBER : 13, 14

NCSS FILENAME/FILETYPE : (13) ETAM/VDATA; (14) ETAM/SVDATA

MEDIA : DASD

FORMAT : F, 80, 80

CREATED/GENERATED BY : Not Applicable

CONTENTS PRINTED BY : Not Applicable

FILE FUNCTION : Input card deck with data for the load of the Abbreviated Vehicle data file

The VDATA card deck is the input file as transmitted to the NCSS time-sharing vendor. This file may be in unsorted order.

The SVDATA file has been sorted into ascending sequence by the Vehicle Stock Number portion of the record. This sort is accomplished by running the executive sequence STARTUP.

Record Position	Data Name	Length	Type	Value/Comments
1-13	Vehicle Stock Number	13	Char	
14-22	Vehicle Designator	9	Char	
23-69	Vehicle Nomenclature	47	Char	
70-79	Vehicle Cost	10	F(10,2)	P'(8)ZVZZ'; leave blank, if unused
80	Filler	1	Char	Blank

FILE REFERENCE NUMBER : 15
NCSS FILENAME/FILETYPE : ETAM/VEHS
MEDIA : DASD
FORMAT : F, 80, 80
CREATED/GENERATED BY : Program P8
CONTENTS PRINTED BY : Program P8A
FILE FUNCTION : Abbreviated Vehicle Data File

Record Position	Data Name	Length	Type	Value/Comments
1-13	Vehicle Stock Number	13	Char	
14-22	Device Designator	9	Char	
23-69	Device Name	47	Char	
70-79	Device Cost	10	F(10,2)	P'(8)9V99' This field currently not available for all records
80	Filler	1	Char	Blank

FILE REFERENCE NUMBER : 16

NCSS FILENAME/FILETYPE : ETAM/DVEH

MEDIA : DASD

FORMAT : F, 750, 750

CREATED/GENERATED BY : Program P8

CONTENTS PRINTED BY : Not Applicable

FILE FUNCTION : Abbreviated Vehicle Directory, ordered by Vehicle Stock Number

Record Position	Data Name	Length	Type	Value/Comments
1-13	Vehicle Stock Number	13	Char	
14-15	Pointer to Relative Record in Abbreviated Vehicle File (File Number 15)	2	Fixed Bin	
16-750	Repeat of Pos (1-15), above, Occurs 50 Times Total	734		Unused elements of the array are blanked (Vehicle Stock Number) or zeroed (pointer)

FILE REFERENCE NUMBER : 17, 18

NCSS FILENAME/FILETYPE : (17) ETAM/TDSCR; (18) ETAM/STDSCR

MEDIA : DASD

FORMAT : F, 80, 80

CREATED/GENERATED BY : Not Applicable

CONTENTS PRINTED BY : Not Applicable

FILE FUNCTION : Input card deck with Descriptors as assigned to Task data

The TDSCR file is the Task Descriptor data as transmitted to the NCSS time-sharing vendor. This file may be in unsorted sequence.

The STDSCR file has been sorted into ascending sequence by the field represented by the first 13 characters of the record. This sort is accomplished by running the executive sequence STARTUP.

Record Position	Data Name	Length	Type	Value/Comments
1-5	Rating	5	Char	
6-7	Rank	2	Char	
8-13	Jobtask Number	6	Char	
14-15	2-Digit Descriptor	2	F(2)	P'99'
16	Filler	1	Char	Blank
17-79	Pos 14-16, Repeated 21 Times	63		
80	Filler	1	Char	Blank

Notes - Max of 22 descriptors per card; if full card not required, leave remainder blank

- Each descriptor is a two-digit number; supply lead zero, if applicable.

FILE REFERENCE NUMBER : 19

NCSS FILENAME/FILETYPE : ETAM/TDESC

MEDIA : DASD

FORMAT : F, 213, 213

CREATED/GENERATED BY : Program P9

CONTENTS PRINTED BY : Program P20 (Option T)

FILE FUNCTION : Descriptor data to match the contents of the Abbreviated Task data file (File Number 22)

Record Position	Data Name	Length	Type	Value/Comments
1-5	Rating	5	Char	
6-7	Rank	2	Char	
8-13	Job Task Number	6	Char	
14-213	Descriptor Flags, Occurs 100 Times	200	Fixed Bin	Initially, all 100 flags are zeroed

Note - Descriptor Flags (subscripted 1 to 100, incl.) are NON-ZERO for those descriptors appliable to the Task keyed by the first 13 characters of the record.

FILE REFERENCE NUMBER : 20, 21

NCSS FILENAME/FILETYPE : (20) ETAM/TDATA; (21) ETAM/STDATA

MEDIA : DASD

FORMAT : F, 80, 80

CREATED/GENERATED BY : Not Applicable

CONTENTS PRINTED BY : Not Applicable

FILE FUNCTION : Input data deck for the load of the Abbreviated Task data file

The TDATA deck is the input data as transmitted to the NCSS time-sharing vendor. This file may be unsorted.

The STDATA file has been sorted into ascending sequence by the field represented by the first 13 characters of the record. This sort is accomplished by running the executive sequence STARTUP.

Record Position	Data Name	Length	Type	Value/Comments
1-5	Rating	5	Char	
6-7	Rank	2	Char	
8-13	Jobtask Number	6	Char	
14-63	Job Description	50	Char	
64-69	Filler	6	Char	Blank
70-79	Billet Cost	10	P(10,2)	P'(8)ZVZZ'; leave blank, if unused
80	Filler	1	An	Blank

FILE REFERENCE NUMBER : 22
NCSS FILENAME/FILETYPE : ETAM/TASKS
MEDIA : DASD
FORMAT : F, 80, 80
CREATED/GENERATED BY : Program P9
CONTENTS PRINTED BY : Program P9A
FILE FUNCTION : Abbreviated Task Data File

Record Position	Data Name	Length	Type	Value/Comments
1-5	Rating	5	Char	
6-7	Rank	2	Char	
8-13	Jobtask Number	6	Char	
14-63	Job Description	50	Char	
64-69	Filler	6	Char	Blank
70-79	Billet Cost	10	F(10,2)	P'(8)9V99'
80	Filler	1	Char	Blank

FILE REFERENCE NUMBER : 23
NCSS FILENAME/FILETYPE : ETAM/DTASK
MEDIA : DASD
FORMAT : F, 1500, 1500
CREATED/GENERATED BY : Program P9
CONTENTS PRINTED BY : Not Applicable
FILE FUNCTION : Abbreviated Task File Directory, ordered by the first 13 characters of the Task File record (includes RATING, RANK, and JOBTASK Number)

Record Position	Data Name	Length	Type	Value/Comments
1-5	Rating	5	Char	
6-7	Rank	2	Char	
8-13	Jobtask Indicator	6	Char	
14-15	Pointer to Relative Record in Abbreviated Task File (File #22)	2	Fixed Bin	
16-1500	Repeat of (1-15), Above, Occurs 100 Times Total	1484		Unused elements of the array are blanked (all char fields) or zeroed (pointer)

FILE REFERENCE NUMBER : 24

NCSS FILENAME/FILETYPE : Not Applicable

MEDIA : TAPE

FORMAT : FB, 810, 3240, DENSITY = 1600BPI, STANDARD LABEL

CREATED/GENERATED BY : Not Applicable

CONTENTS PRINTED BY : Not Applicable

FILE FUNCTION : Abbreviated Course file input is selected from this tape

Record Position	Data Name	Length	Type	Value/Comments
1-4	CDP	4	Char	
5-12	CIN	8	Char	
13-28	Course Short Title	16	Char	
29-32	NOBC	4	Char	
33-36	NEC	4	Char	
37-39	Off. Crs. Code	3	Char	
40-44	Priority Des	5	Char	
45-48	RMS Cost Code	4	Char	
49-50	Type Course	2	Char	
51	SVC Supp	1	Char	
52	Method Inst.	1	Char	
53-67	Filler	15	Char	
68	Dept-Code	1	Char	
69-71	Filler	3	Char	
72	Status-Code	1	Char	
73-75	Status-Date	3	P	S9(5) COMP-3
76-83	Prereq-CIN	8	Char	

Record Position	Data Name	Length	Type	Value/Comments
84-85	Est. Attr. Rate	2	P	S99V9 COMP-3
86-87	Est. Stbk. Rate	3	P	S99V9 COMP-3
88-90	Theory Hrs.	3	P	S9(5) COMP-3
91-93	Lab Hrs.	3	P	S9(5) COMP-3
94-96	Filler	3	Char	
97	TRAPS Ind.	1	Char	
98-102	TPC	5	Char	
103-107	Stu. UIC	5	Char	
108-112	Staff UIC	5	Char	
113-140	CRS Contact Hrs	28		
(7x)	Contact Ratio	2	P	S99 COMP-3
	Contact Hours	2	P	S999 COMP-3
141-143	Total Cont Hrs	3	P	S9(5) COMP-3
144	CFY-Cross Util	1	Char	
145-147	CFY-Crs Length	3	P	S9(5) COMP-3
148-150	CFY-Class Frequency	3	P	
151-153	CFY-Pers Input	3	P	
154-156	CFY-Pers Freq	3	P	
157-159	CFY-Equip Input	3	P	
160-162	CFY-Equip Freq	3	P	
163-165	CFY-Space Input	3	P	
166-168	CFY-Space Freq	3	P	
169-183	Filler	15	Char	
184-223	FY+1 Capacities			same as Col. 144-183
224-263	FY+2 Capacities			same as Co. 144-183
264-286	CFY OF Plan-USN	3	P	S9(5) Comp-3
267-269	CFY OF Plan-USNOB	3	P	
270-272	CFY OF Plan-USNR	3	P	

Record Position	Data Name	Length	Type	Value/Comments
273-275	CFY OF Plan-USNRR	3	P	
276-278	CFY OF Plan-USMC	3	P	
279-281	CFY OF Plan-USCG	3	P	
282-284	CFY OF Plan-USA	3	P	
285-287	CFY OF Plan-USAF	3	P	
288-290	CFY OF Plan-NATG	3	P	
291-293	CFY OF Plan-FORNAT	3	P	
294-296	CFY OF Plan-DOD	3	P	
297-299	CFY OF Plan-NDOD	3	P	
300-302	CFY OF Plan-WOM	3	P	
303-341	CFY EN Plan	39		These fields are all formatted the same as CFY OF PLAN in Col. 264-302 (13.S9(5) COMP-3)
342-380	FY+1 OF Plan	39		
381-419	FY+1 EN Plan	39		
420-458	FY+2 OF Plan	39		
459-497	FY+2 EN Plan	39		
498-536	FY+3 OF Plan	39		
537-575	FY+3 EN Plan	39		
576-614	FY+4 OF Plan	39		
615-653	FY+4 EN Plan	39		
654-692	FY+5 OF Plan	39		
693-731	FY+5 EN Plan	39		
732-770	FY+6 OF Plan	39		
771-809	FY+6 EN Plan	39		
810	Filler			

FILE REFERENCE NUMBER : 25

NCSS FILENAME/FILETYPE : Projectname/LOAD

MEDIA : DASD

FORMAT : F, 80, 80

CREATED/GENERATED BY : Not Applicable

CONTENTS PRINTED BY : Program P1, P2

FILE FUNCTION : Load card file for any/all files of a given project

Record Position	Data Name	Length	Type	Value/Comments

This is the load deck for one or more project files; Program P17 loads the indicated files by reference to control cards present in the load deck. The control cards are:

Initialize Card: If present, must be the first card of the deck. Signals program P17 to load (initialize) all project files (card-image only).

1-4	Control Field	4	Char	"INIT"
5-80	Filler	76	Char	Blank

File Card: Indicates that all cards following, up to the next file card or the end-of-file, are to be loaded to the indicated file.

1-2	Control Field	2	Char	"**"
4-7	Filetype; legal Files are: ID REC REV REJ EXD BPT RKP RRPJ RRPK VQAL VARF SCEN ALTP	4	Char	

FILE REFERENCE NUMBER : 26
NCSS FILENAME/FILETYPE : Projectname/ID
MEDIA : DASD
FORMAT : F, 80, 80
CREATED/GENERATED BY : Program P17
CONTENTS PRINTED BY : Program P17, P1
FILE FUNCTION : Project Identification Information

Record Position	Data Name	Length	Type	Value/Comments
1-80	Descriptive information related to a specific project	80	Char	

All of the records of this file are free-format except for the first card-image; the format for the first card is:

1-1	Project ID	10
11-80	Project Title	70

FILE REFERENCE NUMBER : 27, 28, 29

NCSS FILENAME/FILETYPE : (27) Projectname/REC
(28) Projectname/REV
(29) Projectname/REJ

MEDIA : DASD

FORMAT : F,80,80

CREATED/GENERATED BY : Program P17

CONTENTS PRINTED BY : Program P1 [illegible]

FILE FUNCTION : Search Descriptor Arguments for Courses (REC), Vehicles (REV), and Tasks (REJ)

Record Position	Data Name	Length	Type	Value/Comments
1-10	Descriptor Number	10	Char	1/2 Digit Number, Optionally Signed; If Signed, No Blank Between Sign and Leftmost Digit
11-80	Filler	70	Char	Can Be Used as Comment, If Desired-Not Machine Processable

FILE REFERENCE NUMBER : 30

NCSS FILENAME/FILETYPE : Projectname/SARG

MEDIA : DASD

FORMAT : F,240,240

CREATED/GENERATED BY : Program P2

CONTENTS PRINTED BY : Not Applicable

FILE FUNCTION : Search Argument input to the Range-of-Effects Program P3

Record Position	Data Name	Length	Type	Value/Comments
1-8 below	Record Type	8	Char	See notes,
9-10	Total Number of Active Descriptor Search Arguments	2	Fixed Bin	
11-240	Fixed Bin Array, Occurs 115 Times	230	Fixed Bin	

Notes

- Record type will be indicated by one of the words 'courses', 'vehicles', 'tasks'; there are 3 records in this file, in order as shown.

- For each entity (Courses, Vehicles or Tasks), the numeric array will be zeroed except for those positions which indicate a search designator. The subscript of a non-zero position indicates the designator number. The non-zero contents will be positive for a normal argument, or negative for a "NOT" argument.

FILE REFERENCE NUMBER : 31, 32

NCSS FILENAME/FILETYPE : (31) Projectname/RESU
(32) Projectname/REE

MEDIA : DASD

FORMAT : F, 16, 16

CREATED/GENERATED BY : (RESU) by Program P3
(REE) by Program P5B

CONTENTS PRINTED BY : P1

FILE FUNCTION : RESU is the un-edited output of the Range-of-Effects Search Program P3; REE is the edited output of Program P5B

Record Position	Data Name	Length	Type	Value/Comments
1	Record Type	1	Char	P'9' 1 = Course 2 = Vehicle 3 = Task
2	Record Status	1	Char	P'9' 1 = Original Search Result 2 = Added Via Interactive Program 5B 3 = Deleted Via Program P5B
3	Abbreviated Data Status	1	Char	P'9' 1 = Abbreviated Data is Available 2 = No Data Match in Current Abbr File
4-16	Entity Identifier, As Follows:	13	Char	
	Courses: 4-11 CIN	8		
	12-15 CDP if available	4		
	16 Filler	1		
	Vehicles: 4-16 Veh Stock No.	13		
	Tasks : 4-8 Rating	5		
	9-10 Rank	2		
	11-16 Jobtask Number	6		

FILE REFERENCE NUMBER : 33, 34, 35, 36, 37, 38, 39, 40, 41

NCSS FILENAME/FILETYPE : (33) Projectname/EXD
(34) Projectname/BPT
(35) Projectname/RKP
(36) Projectname/RRPJ
(37) Projectname/RRPK
(38) Projectname/VQAL
(39) Projectname/VARF
(40) Projectname/SCEN
(41) Projectname/ALTP

MEDIA : DASD

FORMAT : F, 80, 80

CREATED/GENERATED BY : Program P17

CONTENTS PRINTED BY : Program P1

FILE FUNCTION : Miscellaneous card-image files; can be initialized by Program P17

None of these files perform a functional task in the Range-of-Effects search portion of ETAM. Their function is explained in previous section of this document that present the system design for those segments of ETAM that are to follow the ROE search.

Since these are card-files, they may initialized by providing proper input to the Project File Initialization Program (P17).

SECTION D.5

MISCELLANEOUS CONTROL CARD SOURCE LISTINGS

Following are listings of the control card sequences used to build the current ETAM Range-of-Effects system at the NCSS time-sharing vendor.

All control decks are NCSS filename = ETAM. They are presented within this section in alphabetic order by filetype. Refer to the File Reference Number as indicated in the following table for the specific format of each control deck.

Filename	Filetype	File Reference Number	Usage
ETAM	CDESC	4	Course Descriptor data, as transmitted to the NCSS time-sharing vendor's computer site
ETAM	COURSE	7	This is the Abbreviated Course File; normally, this is output from Program P14. To conserve time, use of the complete NITRAS MCRF Extract tape and P14 were bypassed
ETAM	DICRDS	2	Control card input for the generation of the Descriptor Master Index File (File Number 3, ETAM/DIND)
ETAM	TDATA	20	Data load for the Abbreviated Task data file (File Reference 22)
ETAM	TDSCR	17	Data load for the Descriptor information associated with Task type entities
ETAM	VCDESC	10	Descriptor load data for entity type Vehicles
ETAM	VDATA	13	Data load for the Abbreviated Vehicle data file (File Reference 15)

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM CDESC

  1   OFFLINE READ  ETAM CDESC
  2   H 2E3710 01 02 03 10 12 13 20 21 30 40 44 46 49 52 64 66 69 72 84 85
  3   H 2E3925 01 02 03 10 12 13 20 22 33 40 43 46 49 52 61 62 63 64 65 66 69 72 85
  4   G 2E6435 01 02 03 12 13 20 21 22 33 40 41 43 44 46 47 49 52 61 62 63 65 72 84
  5   A 2G0014 01 02 03 12 33 41 42 44 45 48 52 53 65 69 82 84
  6   J 2G0535 01 02 03 10 11 12 13 20 21 33 41 44 46 49 52 61 62 63 64 65 72 80 81
  7   J 2G0535 82 85
  8   D 3A0010 01 02 03 12 13 20 21 33 42 45 48 49 53 63 65 82
  9   A 8C0015 01 05 12 20 32 80
 10   C6023536 03 11 12 13 31 40 42 43 45 46 48 49 64 65 68 84
 11   J6440914 01 02 03 04 20 21 32 33 42 45 48 49 53 60 65 66 69 82
 12   E6461641 03 04 05 11 12 13 20 31 40 42 44 45 46 48 49 52 53 62 64 65 80 81 84
 13   C6463103 02 03 04 10 12 31 32 33 80
 14   A6510010 03 11 12 13 31 80 81
 15   A6510020 03 10 11 12 13 20 21 31 64 65 84
 16   A6520050 11 12 13 30 31 40 41 42 43 44 45 46 47 48 49 62 63 64 65 84
 17   A6520068 05 11 12 13 31 40 44 46 64 65 84
 18   J6520474 03 05 11 13 30 71 80
 19   G6526160 01 02 03 12 13 30 33 40 42 43 45 46 48 49 71 80 81
 20   L6610056 03 11 12 13 31 40 43 46 49 52 62 63 64 65 85
 21   A6700011 11 12 13 20 31 61 62 64 68 85
 22   A6700025 05 10 11 12 13 20 32 42 45 48 64 82
 23   F7000010 12 13 20 22 31 40 44 46 49 52 64 66 68 85
 24   A7010027 10 11 13 20 22 31 40 43 46 49 51 61 62 64 66 68 86
 25   N7010320 03 05 10 11 12 13 20 31 40 44 46 49 51 61 62 63 64 85
 26   A7020024 31 40 43 46 49 51 61 62 66 68 86
 27   A7100016 01 02 03 04 10 11 12 13 20 21 32 33 42 45 48 49 53 62 63 65 66 82
 28   A7200013 02 03 11 12 13 31 33 63 64 69 84
 29   A7120011 12 13 20 40 43 46 49 52 60 66 68 86
 30   E 2A1001 11 12 13 30 40 42 43 45 46 48 49 51 53 60 61 64 65 67 70 71 86 87
 31   E 2A1301 01 05 10 11 12 13 20 30 40 42 43 45 46 48 49 51 53 60 61 62 63 64 65 70
 32   E 2A1301 71 86
 33   E 2A1803 01 03 04 10 11 12 13 30 40 42 43 45 46 48 49 51 53 60 61 62 64 65 67 70
 34   E 2A1803 86
 35   E 2A1902 01 02 03 05 10 11 12 13 20 30 42 45 48 49 51 60 61 62 63 64 65 67 71 72
 36   E 2A1902 86 87
 37   C 2A3502 10 13 30 42 45 48 49 53 71 89
 38   C 2A3921 12 13 30 42 45 48 49 53 71 89
 39   C 2A3531 13 30 42 45 48 50 53 80
 40   C 2A3831 01 12 30 42 45 48 50 53 80
 41   D 2C0010 01 03 05 11 12 13 20 30 42 45 48 53 72 80
 42   D 2C0011 01 02 03 11 12 13 30 40 42 43 45 46 48 49 51 53 60 63 64 69 72 80 85
 43   E 2C0012 01 02 03 11 12 13 30 42 45 48 53 72 80
 44   Q 2C0015 03 11 12 13 20 30 40 43 46 49 51 64 67 68 85
 45   E 2C0901 01 02 03 11 12 13 20 30 40 42 43 45 46 48 49 51 53 61 62 63 64 67 69 72
 46   E 2C0901 85 86
 47   C 2C3352 01 03 05 12 30 80
 48   E 2D0016 01 02 03 05 80
 49   E 2D0028 03 11 12 13 20 22 30 41 42 44 45 47 48 53 72 80 8[illegible]
 50   E 2D0032 01 02 03 11 12 13 30 41 42 44 45 47 52 53 72 80 82 84
 51   E 2D0047 01 03 04 10 12 13 20 21 30 42 45 48 53 60 64 65 66 72 82
 52   E 2D0072 01 03 05 12 13 30 42 45 48 53 62 63 65 71 80 84
 53   E 2D0075 01 03 10 11 12 13 20 30 40 44 49 52 62 63 64 82 84
 54   E 2D1302 30 41 44 47 49 52 61 62 63 67 71 85 87
 55   E 2D1601 30 41 44 47 49 52 61 62 63 64 65 67 72 86
 56   A 2E0013 01 02 03 10 21 33 65 80 82
 57   J 2E0100 01 03 05 10 11 12 13 20 21 30 42 45 48 53 62 64 65 71 80 83
 58   K 2E1078 01 02 03 05 10 12 21 33 42 45 49 53 65 66 72 82
 59   B3010033 02 03 04 10 11 12 13 30 32 42 45 46 49 53 61 62 63 64 69 83 84
 60   B3030051 02 03 05 10 11 12 13 20 22 30 32 40 42 43 45 46 48 49 52 62 63 64 66
 61   B3030051 68 69 84
 62   A4120010 01 02 03 05 11 12 13 20 22 30 32 40 42 43 45 46 48 49 53 62 63 64 66
 63   A4120010 80 81 82 83 84
 64   C4202011 03 05 11 12 13 20 21 30 32 42 45 48 49 53 63 64 65 83
 65   A 4A0025 03 12 21 22 33 80
 66   A4310011 01 03 05 10 11 12 13 20 21 22 30 40 41 43 44 46 47 49 52 60 61 62 63
 67   A4310011 64 66 68 84
 68   A4310014 01 03 13 20 22 30 64 65 66 72 86 89
```

```
PAGE NUMBER :    2

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM CDESC

 69   A4330019 04 30 40 43 46 49 51 62 63 64 68 86 87 89
 70   A4330025 03 12 13 20 30 40 42 43 45 46 48 49 51 61 62 63 64 66 68 72 87 88
 71   A4910014 01 02 03 04 10 11 12 13 20 21 30 40 42 44 45 46 47 49 52 60 61 62 63
 72   A4910014 64 66 72 84
 73   A5000011 01 03 12 13 21 32 42 45 49 53 63 65 69 72 84
 74   A5000028 01 03 04 11 12 13 20 32 42 45 53 62 64 65 83
 75   S5000029 01 02 03 04 10 12 13 20 32 44 60 65 66 84
 76   A5000032 01 02 03 04 10 13 20 21 32 60 65 66 72 83
 77   A5100012 01 03 04 11 12 13 20 32 40 44 46 49 52 63 64 68 69 85
 78   C5163203 03 04 11 12 13 20 32 42 45 48 49 52 64 84
 79   A5320015 01 03 11 12 13 20 22 32 63 64 66 85
 80   A5420014 03 11 12 20 32 64 83
 81   A5510019 03 20 30 40 42 43 45 46 48 51 53 64 67 68 84
 82   A5510027 05 11 12 13 20 31 64 68 83 84
 83   A5510068 11 13 20 30 40 43 46 49 51 64 67 68 85
 84   A5700010 03 10 11 13 20 22 32 40 43 46 49 51 63 64 66 69 85
 85   A5800016 02 03 11 12 13 20 30 32 40 43 46 49 51 63 64 68 72 84
 86   E6000506 02 03 04 11 12 13 20 30 41 42 44 45 47 48 52 61 62 63 64 68 84
 87   C6003358 11 12 13 20 31 42 45 48 53 64 81
 88   C6003472 01 03 11 13 31 80
 89   C6003831 11 12 13 20 31 42 45 48 53 64 80
 90   E6010210 01 03 11 12 13 20 31 40 42 43 45 46 48 51 53 64 83 85
 91   C6023353 05 11 12 13 31 42 45 48 53 64 80 81
 92   C6023482 03 11 12 13 31 42 45 48 53 64 80 81
 93   J2330211 01 02 03 04 05 12 13 20 33 42 45 53 63 65 82
 94   A2420010 05 20 21 22 30 32 63 64 65 66 85
 95   J2430974 03 05 10 11 12 13 20 30 40 44 49 52 62 64 84
 96   A2440010 05 10 11 12 13 32 60 62 63 66 80
 97   J2500313 01 02 03 04 10 11 30 42 45 48 65 66 80
 98   83000010 02 03 05 10 11 12 13 20 30 62 63 64 65 81 84
 99   83000012 03 04 05 11 12 13 20 21 30 32 62 63 64 65 69 81 83
100   J1300645 01 03 11 12 30 40 42 45 64 80 81 82
101   J8000433 12 20 30 40 42 44 46 49 52 64 66 85
102   J 3A9512 01 03 04 11 12 13 20 22 30 42 45 49 53 62 63 66 82
103   P 003308 01 02 03 04 11 12 20 21 22 33 42 45 48 53 65 66 82 83
104   K 002044 01 03 04 11 12 13 20 22 32 64 65 66 82 84
105   A 000111 01 03 04 11 12 13 20 21 22 33 42 45 48 53 65 66 82
106   C 003491 01 03 11 12 13 30 42 45 62 63 80
107   H 00521C 01 02 03 04 21 33 65 66 80 82 83 89
108   H 005320 03 12 13 30 41 42 44 45 49 52 53 64 68 84
109   C 003807 01 03 11 12 30 33 42 45 48 53 60 80
110   C 003722 01 03 30 33 80
111   A1020060 03 31 64 65 71 85
112   A1020093 03
113   A1010108 30 31 71 80 83
114   L1010024 05 80
115   N1000555 05 11 12 13 20 21 31 42 45 48 53 64 65 71 84
116   C1003834 01 03 05 11 12 13 31 64 65 83 84
117   A1020095 05 11 12 13 20 21 31 64 65 82 83
118   C1023793 01 05 11 12 13 31 64 80 81 83 84
119   C1213011 01 05 10 11 12 13 30 31 40 42 44 45 46 48 64 65 80 81 84
120   A1930050 05 31 80
121   M198100E 03 11 20 31 40 44 46 49 52 64 65 84 85
122   R2106104 05 20 32 42 45 48 64 83 89
123   K2210042 01 05 20 30 42 45 53 63 64 82 83
124   J2210348 03 10 12 13 30 41 44 49 53 61 62 64 65 87 89
125   J2210344 02 03 04 12 13 20 30 41 42 44 45 47 48 49 53 61 62 63 64 69 85
126   J2210357 01 02 03 04 30 33 62 64 65 66 69 72 83 84
127   K2220035 02 03 20 30 41 44 49 52 61 62 64 67 69 72 86
128   C2222010 01 02 03 04 11 12 13 20 30 41 42 44 45 49 53 61 62 63 64 67 80 81 82
129   C2222010 83
130   J2210303 03 05 12 30 33 42 45 48 50 62 63 64 80
131   J2210310 01 02 03 04 13 30 41 44 49 52 61 62 63 64 67 69 72 85
132   K2330066 01 02 03 05 13 14 20 30 40 41 44 52 61 62 63 64 81 84
133   J2330203 03 05 10 11 12 13 20 30 40 44 49 52 61 62 63 64 84 85
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM COURSE

 1   OFFLINE READ  ETAM COURSE
 2   0327A 000111SWO ADV COMMAND     0000  PDBC21LA00000T0435000100104022401600000000
 3   0303A 2E0013TAL PCD/PXO ORI 0000000012KBAC21PA62171 03250000000012080000000000000
 4   0143A 2G0014MINE C/M OFF    9268000025CCBC21LA72183 04311000000033120030000000000
 5   2131A 8C0015WAREHOUSE OPS   0000000024FAAF21L 63203 06140005005005035000000000000
 6   516UA1010108WRA-4 CMB MA        000002MBDF11L 00000 41422050000005021009000000000
 7   5535A1010108WRA-4 CMB /FINIS    000001FCAF11  00000 04111070000012000000000000000
 8   7654A1010108WRA-4 CKB MA    0000000002BBDF11LA75191 41423050000012034024000000000
 9   2213A1930050INERT NAV PRINC     000007FCAC11  00000 41421000000019000000000000000
10   6529A24200101S A                000014MBBA12L 00000 06110100000072000000000000000
11   6078A4120010EA-A                000026ACBA11  00000 03350070030081000000000000000
12   6287A4120010EA A            0000000026MCBA11L 00000 03350070030081000000000000000
13   1035A4310011EOD BASIC NAVY  9230533225BBBC11LA74182 04380320260269398750000000000
14   1036A4310014EOD REFR NAVY   0000000023BBEC11LA74182 04380010050047092075000000000
15   2040A4330019MASTER DIVQUAL      534125ABJC11LA70181 04370200005036114068000000000
16   2082A4330025DIVER FIRST         534223ABKC11LA70181 04370030005120514141000000000
17   204GA4910014RADIUM REMOV OPS00000000     F11L 00000 00152000000005024008000000000
18   3192A5000011CIAC                958920DFCC11L 00000 06130130020026000000000000000
19   3193A5000011CIAC            0000958920ECDC11LA70135 06130130020024056058000000000
20   2398A50000283M COORD        0000951213DEAC11PA74189 03330000000005000000000000000
21   461WA50000283M COORD        0000951213   C11L 00000 03330000000005000000000000000
22   2010A5000032OVRSEA DIPL COOR0000000029UCDF11LA74120 06210010000005018017000000000
23   202MA5000032OVRSEA DIPL COCR00000000  KBBF11  00000 06210050000008000000000000000
24   2601A5000032OSEAS DIPL COORD00000000  DAAF11  00000 06210000000005035000000000000
25   539JA5000032OVRSEA DIPL COOR    0000  QBHF11LA74251 06210030000010025015000000000
26   6057A5100012YN A                000023SCDA11PA74252T06130190500048000210000000000
27   470PA5320015COBOL PROGRAMING0000274221   C14P 00000 03140020020033000000000000000
28   4717A5420014DK C RETURNS        000026DGBF11LA71305 06140100005012030030000000000
29   0133A5510019CAR HAND BAS        000031DBKF11L 00000 03320000000005020020000000000
30   3931A5510027UNREP MECH/HYD      429505DBDC11L 00000 03320005000026112348000000000
31   0134A5510068FORK LIFT OPER      000031DBPF11L 00000 03320000000003004016000000000
32   3249A5700010BROADCAST SPEC      322126   C18   00000 06130100000007500000000000000
33   6053A5800016CTO A               48MD17QCBA31PA47305 03400170500117240270000000000
34   6069A6510010BT A       /FINIS   000004JGAA11P 00000 03310100100019000000000000000
35   6260A6510010BT A            0000000004JGAA11P 00000 03310100100053000000000000000
36   4138A6510020BT 1200 PSI     0000452305YBDC11BA74182 03310030060054123175000000000
37   5399A65100201200 PSI BT         452307PCEC11L 00000 04350001001061136109000000000
38   5224A65200500X GENR 6L160P  0000425212FBAC11LA73226 41441050000019038049000000000
39   463EA6520068SOLAR GAS TURBIN0000000007JEAC11LA75279T03340100000019038053000000000
40   3078A6700011WATCH REPAIR        181207EDAC11  00000T03112030000082000420000000000
41   2121A6700025PREC PHY MEAS       182107MBDC12L 00000 03110020010078122208000000000
42   3202A7010027WELD/HPRES PIPE 0000495519BDEC11LA75033 03320150020068028290000000000
43   337MA7010027WELD/HPRES PIPE     495519    C11   00000 03320150020054000000000000000
44   351JA7020024MR/C/PH3/PRCGRND0000000023BBGC11LA76085 03320120100033058103000000000
45   3205A7100016GRPVIII ADV MGMT    000026EEAC11LA75136 03350020010040198000000000000
46   346FA7120011BU MILLWORKER       590425ABDC11  00000 03350100010061000000000000000
47   351QA7120011BU MILLWORKER   0000590425MBDC11L 00000 03350100010061000000000000000
48   4028A7200013UT-J                000027CCCC71  00000 03350090010093000000000000000
49   466ZA7200013UT-J            0000000027NBCC71L 00000 03350090010093000000000000000
50   9750C 003491C2A A/C FAM     0000000005NSAF21LA73032 04220030000010053000000000000
51   2887C 003722TA4JF FAM PILOTS0000000005NEAF21LA76114 04220030000003016000000000000
52   9007C 003722TA4JF FAM PILOTS    000005NMAF21LA73305 04220030000003016000000000000
53   9009C 003722TA4JF FAM PILOTS    000005NJAF24LA73305 04220030000003016000000000000
54   9010C 003722TA4JF FAM PILOTS    000005NRAF21LA73305 04220030000003016000000000000
55   9738C 003722TA4JF FAM PILOTS0000000005NZAF21LA73305 04220030000003016000000000000
56   9033C 003807F4J AWG10 MSL       000001NFAF24LA74121 04220030000004020007000000000
57   9051C 2A3502KC130F PILOT REF    000005    F24LA72092 04220030000002014000000000000
58   9751C 2A3531P3 FAM PILOTS   0000000001NKAF21LA72306 04220030000004027000000000000
59   2374C 2A3831F/RF4B/J/N FAM  0000000001NWAF21LA74121 04220030000005027000000000000
60   540MC 2A3831F/RF4B/N/J FAM  0000000001NSAF21LA74121 04220030000005027000000000000
61   540NC 2A3831F/RF4B/N/J FAM  0000000001NFAF24LA74121 04220030000005027000000000000
62   9062C 2A3831F/RE-4B/N/J FAM     000001NIAF24LA74121 04220030000005027000000000000
63   2576C 2C3352UH1N A/C FAM/P/ 0000000006NIAF24LA72275 04220030000004027000000000000
64   7529C1003834AAS18 INT MAI       000001NIAC14LA71091 04220030000026133000000000000
65   7914C1023793A7AB ATT HEADING    710501NDAC11LA74060 04220030000010027023000000000
66   7915C1023793A7AB ATT HEADING    710501NPAC11LA74060 04220030000010027023000000000
67   346LC1213011AWM23 RADIO FREQ    798401NSAC11LA75329 04220030000129152282000000000
68   544LC1213011AWM23 RADIO FREQ0000798401NWAC11LA75345 04220030000089152282000000000
```

```
PAGE NUMBER :    2

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM COURSE

 69   6278C2222010AC A1           0000692212FBBA11PA75056T0425015025009619622200000000
 70   4510C4202011MOAF C7             741221PBCC71L 00000 0421010002012364006000000000
 71   2717C5163203AVI MAI MATERIAL0000000001N4AF11 A75260 0422003000000403200000000000
 72   321XC5163203AVI MAI MATERIAL    000004NPAF11LA75244 0422003000000502800000000000
 73   321ZC5163203AVI MAI MATERIAL    000004N2AF11LA75244 0422003000000502800000000000
 74   322BC5163203AVI MAI MATERIAL    000004NDAF11LA75244 0422003000000502800000000000
 75   322DC5163203AVI MAI MATERIAL    000004NUAF11LA75244 0422003000000502800000000000
 76   322EC5163203AVI MAI MATERIAL    000004NVAF11LA75244 0422003000000502800000000000
 77   4964C6003358PWR TRN ROTORS      000001   F14  00000 0422000000001200000000000000
 78   340CC6003472E2C WEA SYS FAM     000001NUAF11LA75152 0422003000000503400000000000
 79   7540C6003831RF4 AC MECH MAI     000001NFAC14LA75060 0422003000001707501200000000
 80   7541C6003831RF4AC MECH MAI      000001NIAC14LA75060 0422003000001707501200000000
 81   7938C6023353AH1J ELEC ORG       000001NIAF14LA72306 0422003000001203902800000000
 82   204EC6023482E2C VAPOR CYCLE 0000000001NUAF11LA74335 0422003000000503400000000000
 83   2667C6023482E2C VAPOR CYCLE 0000000001NSAF11LA74335 0422003000000503400000000000
 84   2671C602353654H6077 PROP INT0000000001NKAF11LA74335 0422003000001002202800000000
 85   341DC602353654H6077 PROP INT    000001NTAF11LA74335 0422003000001002202800000000
 86   349HC6463103CVA/CV ALW SUPV 0000000001NVAF11LA74152 0422003000001205900700000000
 87   349KC6463103CVA/CV ALW SUPV 0000000001NUAF11LA74152 0422003000001205900700000000
 88   7365C6463103GVA/CV ALW SUPV     000001NCAF11LA74152 0422003000001205900700000000
 89   9279J 2E0100WEAPONS DEPT OFF0000000022GEBC21LA70203 TLANB00000002609303400000000
 90   528EJ2210303ECM/ECCM        0000000020ECCF11LA72213 TLANB00000000502401100000000
 91   534AJ2210303JAM RECOG & ECM     000010DBCF11LA73222 TLANG00000000503000700000000
 92   538PJ2210310CIC TEAM TRAIN  0000000010ECDF11LA72192 TLANB00000000500602900000000
 93   526KJ2210344NAVIG CIC TM TRG0000000013YHHF11LA74070TTLANT03006000300701400000000
 94   528HJ2210344RADNAV TEAM TRNG0000000016ECGF11LA70187 TLANB00000000500802700000000
 95   528JJ2210348CATCC TEAM TRNG 0000000020ECJF11LA71187 TLANB00000000500203300000000
 96   533PJ2210357CIC TEAM TRNG   0000000005YHGF11LA74080TTLANT03006000300401400000000
 97   538QJ2210357MULTI-THREAT TNG0000000014ECQF11LA70194 TLANB00000000500602900000000
 98   2637J2330203ADV EW OP'S CRSE00000000  DNAF11LA75321       00000000500000000000000
 99   205WJ2330211SURFACE EW SUPVR0000000007YEJF11LA75182TTLANT03006001004101500000000
100   525UJ2330211SURFACE EW SUPVR0000000014FCCF11LA75174 TLANB05000001004100700000000
101   5324J2330211SURF EW SUP         000012AAAF11  00000 TLANH00000000805600000000000
102   5332J2330211SURF EW SUPVR       000010DBCF11LA72183 TLANG00000000100400080000000
103   542NJ2330211SURFACE EW SUPVR0000000021000F21 A75135 TLANT03000001203003000000000
104   2181J2430974IPC                 000007LBMF11  00000 TLAND00000000501701600000000
105   265VJ2500313DD963 OP/WEP CON00000000  DFBF11 P76082       00000000500000000000000
106   266CJ2500313DD963 OP/WEP INT0000000007ZAAF1  A76082 TLANT03006000503000000000000
107   1034J6440914MAUW NUWEAPS        077122500C11LA75303 TLAN 01000501206001000000000
108   8178J6520474EN BASIC-NONMAG     000023FBCC11LA72183 TLANG00000001908600400000000
109   9740K 2E1078COORD ASW PT II 0000000012TBZF21LA75209 P002203000000502101400000000
110   5599K2210042ASIS-LCC        0000000032YLJC11LA75001TP002103006001903803200000000
111   534ZK2220035CATCC NTDS      0000000009YHJF11LA73182TP002103006001202005000000000
112   205YK2330066SUB EW OP-ADV   0000000007YEMF11LA75182TP021103006000501901600000000
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM DICRDS

 1   OFFLINE READ  ETAM DICRDS
 2   -- COURSES
 3   ** REFERENCE KNOWLEDGE
 4   01 SYSTEM PURPOSES
 5   02 ORGANIZATIONAL ROLES
 6   03 CONTEXTS OF OPERATION
 7   04 ORGANIZATIONAL RULES
 8   05 OTHER
 9   ** ENABLING KNOWLEDGE
10   10 OPERATIONAL GOAL CRITERIA
11   11 NOMENCLATURE, IDENTIFICATION, LOCATION
12   12 PROCEDURAL DESCRIPTIONS
13   13 JOB RELEVANT FACTS,RULES
14   14 OTHER
15   ** TASK FORMATS
16   20 PROCEDURE FORMATS
17   21 DECISION FORMATS
18   22 CONSTRUCTION FORMATS
19   23 OTHER
20   ** GROSS JOB CATEGORIES
21   30 OPERATIONS
22   31 MAINTENANCE
23   32 SERVICE & ADMINISTRATION
24   33 COMMAND
25   34 OTHER
26   ** OBJECTIVE TASK VARIABLES AS MANIFEST IN THE TRAINING
27   40 EQUIPMENT & OBJECTS USED: REAL
28   41 EQUIPMENT & OBJECTS USED: SIMULATED
29   42 EQUIPMENT & OBJECTS USED: SYMBOLIC
30   43 ENVIRONMENTS IN WHICH TASK IS TRAINED: REAL
31   44 ENVIRONMENTS IN WHICH TASK IS TRAINED: SIMULATED
32   45 ENVIRONMENTS IN WHICH TASK IS TRAINED: SYMBOLIC
33   46 TOOLS/INSTRUMENTS USED IN TRAINING: REAL
34   47 TOOLS/INSTRUMENTS USED IN TRAINING: SIMULATED
35   48 TOOLS/INSTRUMENTS USED IN TRAINING: SYMBOLIC
36   49 REF/ENABLING INFO IN DOING TASK: APPLIED
37   50 REF/ENABLING INFO IN DOING TASK: NOT APPLIED
38   51 CRITERIA OF TASK PERFORMANCE: REAL
39   52 CRITERIA OF TASK PERFORMANCE: SIMULATED
40   53 CRITERIA OF TASK PERFORMANCE: SYMBOLIC
41   ** TASK FUNCTIONS DOMINANT IN TRAINING
42   60 GOAL PROJECTION
43   61 SCAN-DETECT
44   62 IDENTIFY
45   63 INTERPRET
46   64 PROCEDURE FOLLOWING
47   65 DECIDE
48   66 CONSTRUCT, PLAN
49   67 TRACK
50   68 MOTOR PERFORMANCE
51   69 INTERPERSONAL INTERACTION
52   70 RECALL TASK-CYCLE INFORMATION
53   71 RECALL ENABLING INFORMATION
54   72 ADAPT IMPROVISATIONALLY/IMPROMPTU
55   73 OTHER
56   ** STAGE OF LEARNING
57   80 ORIENTATION, FAMILIARIZATION
58   81 TASK NOMENCLATURE, IDENTS, LOCATIONS, FACTS, RULES
59   82 TASK FORMATS AT CONCEPTUAL LEVEL
60   83 PROCEDURES AT VERBAL LEVEL ONLY
61   84 TASK COMPONENTS WITH GUIDANCE
62   85 ENTIRE JOB-TASK PROCEDURALLY AT BARELY ACCEPTABLE MASTERY
63   86 HIGHLY PROFICIENT IN WORK CONTEXT
64   87 UNUSUAL TASK CONDITIONS
65   88 PERFORMANCE AT KEY MAN LEVEL
66   89 REFRESHER LEARNING
67   -- VEHICLES
68   ** VEHICLE TYPES
```

```
PAGE NUMBER :    2

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM DICRDS

 69    01 INSTRUCTOR
 70    02 STATIC GRAPHICS
 71    03 ANIMATED GRAPHICS
 72    04 AUDIO
 73    05 PHYSICAL MODELS
 74    06 PROCEDURAL TRAINERS: SYMBOLIC
 75    07 PROCEDURAL TRAINERS: PHYSICAL BUT NON-FUNCTIONAL
 76    08 PROCEDURAL TRAINERS: FUNCTIONAL
 77    09 TASK & SYSTEM SIMULATORS
 78    10 REAL EQUIPMENT ITSELF
 79    ** CLASS OF TRAINING OBJECTIVE
 80    11 REFERENCE KNOWLEDGE
 81    12 KNOWLEDGE, TASK SPECIFIC/ENABLING
 82    13 TASK-SKILL FORMATS
 83    14 SKILL TRAINING
 84    ** VEHICLE PROPERTIES
 85    21 VISUAL
 86    22 AUDITORY
 87    23 KINESTHETIC/VESTIBULAR
 88    24 TACTILE
 89    ** TYPE OF CONTENT DISPLAYED
 90    31 TEXT-VERBAL
 91    32 DIAGRAMMATIC
 92    33 ABSTRACTED PICTORIAL REPRESENTATION
 93    34 PICTORIAL REPRESENTATIONS
 94    35 PHYSICAL REPRESENTATIONS
 95    36 OTHER
 96    ** TYPE OF PRESENTATIONAL SEQUENCE
 97    41 LIBRARY OF FRAMES OR ITEMS
 98    42 PRESENTATION SEQUENCE NOT APPLICABLE
 99    43 FIXED SEQUENTIAL FRAMES OR ITEMS
100    44 RANDOM SELECTION OF FRAME SEQUENCES
101    45 DYNAMIC CHANGE OF CONTENT WITHIN FRAME
102    ** SELECTION SOURCE FOR SEQUENCING
103    51 INTERNAL PROGRAM
104    52 INSTRUCTOR
105    53 STUDENT CHOICE
106    54 STUDENT PERFORMANCE
107    55 COMBINATIONS OF THE ABOVE
108    ** TYPE OF EXTERNAL CONTROL OPERATED BY STUDENT
109    61 NOT APPLICABLE DIRECTLY
110    62 ARTIFICIAL OR SYMBOLIC RESPONSE
111    63 REPRESENTATIONAL RESPONSE BY SYMBOLIC SELECTION
112    64 REPRESENTATIONAL RESPONSE BY DUMMY CONTROL ACTIVATION
113    65 TASK-MANIPULATIVE RESPONSE, NON-DYNAMIC IN TIME AND FORCE
114    66 TASK-MANIPULATIVE RESPONSE, DYNAMIC AND INTERACTIVE
115    ** FEEDBACK PRESENTATION LOGIC
116    71 NOT APPLICABLE
117    72 SELECTS NEXT STIMULUS ITEM OR SEQUENCE
118    73 GIVES EVALUATION OF PRECEDING RESPONSE
119    74 SELECTS AND PRESENTS GUIDANCE INFO
120    ** RESPONSE EVALUATION LOGIC
121    81 NOT INTERNAL-DEPENDS ON INSTRUCTOR OR STUDENT EVALUATION
122    82 EVALUATION LIMITED TO STUDENT'S IMMEDIATE RESPONSE
123    83 EVALUATION EXTENDED TO A SET OF STUDENT RESPONSES
124    84 TOLERANCE LIMITS ON ACCEPTABLE STUDENT RESPONSE: FIXED
125    85 TOLERANCE LIMITS ON ACCEPTABLE STUDENT RESPONSE: VARIABLE
126    -- TASKS
127    ** ADMINISTRATIVE ROUTINE PAPERWORK
128    01 FORMS FILLING
129    02 DOCUMENT-FILE MANAGEMENT
130    03 DECODE-ENCODE
131    04 SCREEN-FILTER DISTRIBUTE
132    05 OTHER
133    ** ADMINISTRATIVE NON-ROUTINE PAPERWORK
134    11 CONSTRUCT MESSAGES-REPORTS
135    12 ANALYZE-INTERPRET
136    13 CONSTRUCT RECOMMENDATION-PROPOSAL
137    14 CONSTRUCT PLAN
```

```
    PAGE NUMBER :    3

    SOURCE CARD LISTING FOR
    OFFLINE READ  ETAM DICRDS

138    15 OTHER
139    ** ADMINISTRATIVE OFFICE EQUIPMENT OPERATION
140    21 TYPEWRITER
141    22 TELEPHONE,ETC.
142    23 REPRODUCER
143    24 COMPUTER TERMINAL
144    25 TELETYPE
145    26 OTHER
146    ** INTERPERSONAL: INFORM-INSTRUCT-MANAGE
147    31 BRIEF-DEBRIEF
148    32 INSTRUCT-TRAIN
149    33 ASSIGN, MONITOR, COORDINATE
150    34 EVALUATE
151    35 OTHER
152    ** TECHNICAL PROCEDURES
153    41 SEQUENTIAL
154    42 STRATEGIC, ADAPTIVE
155    43 INTERPERSONAL,TEAM MEMBER
156    44 OTHER
157    ** TECHNICAL TYPE OF PROCEDURE(MAIN EMPHASIS)
158    51 SCAN-DETECT: SYMBOLIC (INCL. MAPS,RADAR,TRANSDUCED SIGNALS)
159    52 SCAN-DETECT: NATURAL
160    53 IDENTIFY: SYMBOLIC(INCL. TRANSDUCED SIGNALS)
161    54 IDENTIFY: NATURAL
162    55 INTERPRET:SYMBOLIC
163    56 INTERPRET: NATURAL
164    57 PERCEPTUAL-MOTOR
165    58 COGNITIVE OPERATIONS
166    59 MANUAL
167    60 COMMUNICATE
168    ** TECHNICAL WITH OR WITHOUT EQUIPMENT
169    61 WITH EQUIPMENT (PAPER IS EQUIPMENT)
170    62 WITHOUT EQUIPMENT
171    ** TECHNICAL DECIDE
172    71 DIAGNOSE-ANALYZE
173    72 SELECT-CHOOSE
174    73 UNDER STRESS OR LOAD
175    74 OTHER
176    ** TECHNICAL CONSTRUCT-REPAIR, PLAN
177    81 MANUAL CONSTRUCT OR REPAIR
178    82 COGNITIVE CONSTRUCT, PLAN
179    ** TECHNICAL TRACK-AIM-STEER
180    91 APPLICABLE
181    92 TIME STRESS
182    93 INFORMATION-LOAD STRESS
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDATA

  1   OFFLINE READ  ETAM TDATA
  2   SM     3 34009 REEVE AND SPLICE HALYARDS
  3   SM     3 42020 RENDER PASSING AND SIDE HONORS
  4   SM     3 42021 STAND SIGNAL OPERATOR WATCHES
  5   SM     3 50303 USE MAINTENANCE REQUIREMENT CARDS (MRC)
  6   SM     3 50928 COMPLETE MAINTENANCE DATA FORMS
  7   SM     3 54722 INVENTORY INSTALLED EQUIPMENT
  8   SM     3 72001 PERFORM ROUTINE MAINTENANCE ON OPTICAL EQUIPMENT
  9   SM     3 77264 SECURITY OF COMMUNICATIONS CLASSIFIED MATERIAL
 10   SM     3 84225 OBSERVE COMM. TRANSMISSION SECURITY REQUIREMENTS
 11   SM     3 86015 RECOGNIZE, USE AND INTERPRET EMERGENCY SIGNALS
 12   SM     3 86016 TRANSMIT, RECEIVE FLASHING LIGHT CODE AT 6 GPM
 13   SM     3 86017 TRANSMIT, RECEIVE FLASHING LIGHT PLAIN LANGUAGE
 14   SM     3 86018 TRANSMIT, RECEIVE SEMAPHORE PLAIN LANGUAGE
 15   SM     3 86021 DISTINGUISH FLAGHOIST SIGNALS
 16   SM     3 86022 SELECT AND DISPLAY FLAGS AND PENNANTS
 17   SM     3 86023 EXERCISE RULES OF VISUAL RESPONSIBILITY
 18   SM     3 86024 DIRECT A LIFEBOAT DURING MAN OVERBOARD EXERCISE
 19   SM     3 86025 IDENTIFY FLAGS AND ENSIGNS OF MARITIME NATIONS
 20   SM     3 86031 PREPARE A VISUAL MESSAGE FOR TRANSMISSION
 21   SM     3 86035 RECEIVE, SEND MESSAGES ACCORDING TO PRECEDENCE
 22   SM     3 86036 PERFORM CHALLENGE REPLY PROCEDURES
 23   SM     3 86038 SIGNALMAN DUTIES AS A MEMBER OF A BOATCREW
 24   SM     3 86365 OPERATE ELECTRICAL AND ELECTRONIC VISUAL SIGNALS
 25   SM     3 86366 OPERATE INFRARED AND FLASHING LIGHT EQUIPMENT
 26   SM     3 94530 ROUTINE UPKEEP OF VISUAL SIGNALING EQUIPMENT
 27   YN     3 38605 FILE RECORDS AND CORRESPONDENCE
 28   YN     3 38606 ORDER AND MAINTAIN STOCK OF OFFICE SUPPLIES
 29   YN     3 38822 PERFORM AS MAST YEOMAN
 30   YN     3 38823 OPTICAL CHARACTER RECOGNITION INPUT DOCUMENTS
 31   YN     3 38824 PERFORM RECEPTIONIST DUTIES
 32   YN     3 38825 OPERATE DUPLICATING EQUIPMENT
 33   YN     3 38826 TYPE AT 30 WORDS PER MINUTE
 34   YN     3 38827 TRANSCRIBE OFFICER MESSAGE ORDERS
 35   YN     3 38830 LEAVE REGULATIONS, PREPARE LEAVE AUTHORIZATIONS
 36   YN     3 38831 TYPE CORRESPONDENCE AND MESSAGES
 37   YN     3 38834 ADMINISTRATIVE, OPERATIONAL RECORDS AND REPORTS
 38   YN     3 38835 INITIATE JOB ORDERS AND WORK REQUESTS
 39   YN     3 38843 MAINTAIN OFFICER, WARRANT OFFICER SERVICE RECORDS
 40   YN     3 38852 TYPE INSTRUCTIONS AND NOTICES
 41   YN     3 38958 EFFECT OFFICER RECEIPTS AND TRANSFERS
 42   YN     3 46358 USE AND MAINTAIN PUBLICATIONS AND DIRECTIVES
 43   YN     3 77269 ENSURE SECURITY OF COMMUNICATIONS
 44   YN     3 77284 CARRY OUT MARKING OF CLASSIFIED MATERIALS
 45   YN     3 94620 MINOR MECHANICAL MAINTENANCE, OFFICE EQUIPMENT
 46   YN     2 25269 IDENTIFY FLEET ADMIN., TACTICAL CHAIN OF COMMAND
 47   YN     2 38619 ORGANIZE AND MONITOR MAINTENANCE OF FILES
 48   YN     2 38620 HANDLE OBSOLETE FILES AND RECORDS
 49   YN     2 38627 MAINTAIN A REPORTS CONTROL SYSTEM
 50   YN     2 38836 SUMMARY AND SPECIAL COURTS-MARTIAL RECORDS
 51   YN     2 38840 TYPE AT 40 WORDS PER MINUTE
 52   YN     2 38841 INSTRUCT SUBORDINATES IN OPS DUPLICATING EQUIPMENT
 53   YN     2 38842 MAINTAIN OFFICER PERSONNEL DIARIES
 54   YN     2 38844 ODCR, OTHER REPORTS AS REQUIRED BY MAPMISMAN
 55   YN     2 38846 INTERPRET TRAVEL REGULATIONS
 56   YN     2 38848 NAVY OFFICER CLASSIFICATION SYSTEM CODE STRUCTURE
 57   YN     2 38849 RECOMMENDATIONS FOR DECORATIONS AND AWARDS
 58   YN     2 38850 PREPARE TEMADD ORDERS FOR OFFICER PERSONNEL
 59   YN     2 38851 CARRY OUT OFFICER SEPARATION PROCEDURES
 60   YN     2 38853 ROUTE CORRESPONDENCE AND CLASSIFIED MATERIAL
 61   YN     2 38854 HANDLE OFFICIAL AND REGISTERED MAIL
 62   YN     2 38859 PREPARE FORMS FOR PERSONNEL SECURITY CLEARANCES
 63   YN     2 38957 MAINTAIN TICKLER AND CROSS-REFERENCE FILES
 64   YN     2 38959 PREPARE PAY AND PERSONNEL ACTION DOCUMENTS
 65   YN     2 46360 ORDER PUBLICATIONS AND CHANGES
 66   YN     2 65015 ADVISE AVAILABLE ACTIVITY LOCATIONS, TYPES OF DUTY
 67   YN     2 65500 ADVISE PERSONNEL IN RE PERSONAL SERVICES CENTERS
 68   YN     2 65501 ADVISE PCS MOVE BENEFITS OF NAVY SPONSOR PROGRAM
```

```
PAGE NUMBER :    2

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDATA

 69     YN     1 38629 APPLY PRINCIPLES OF WORK SIMPLIFICATION
 70     YN     1 38630 MONITOR THE REPORTS CONTROL SYSTEM
 71     YN     1 38631 PREPARE MANPOWER AUTHORIZATION CHANGE REQUESTS
 72     YN     1 38634 REVIEW COMPLETED JOB ORDERS AND WORK REQUESTS
 73     YN     1 38855 PAPERWORK FOR INFORMAL/FORMAL INVESTIGATIONS
 74     YN     1 38856 TYPE AT 50 WORDS PER MINUTE
 75     YN     1 38858 CARRY OUT CASUALTY REPORTING, ASSISTING PROCEDURES
 76     YN     1 38860 PAPERWORK FOR SUMMARY AND SPECIAL COURTS-MARTIAL
 77     YN     1 38960 PREPARE CORRESPONDENCE AND MESSAGES
 78     YN     1 38961 DRAFT INSTRUCTIONS AND NOTICES
 79     YN     1 46361 SUPERVISE ALL ASPECTS OF PUBLICATION HANDLING
 80     YN     1 65480 ASSIST THE CASUALTY ASSISTANCE CALLS OFFICER
 81     YN     1 77285 MONITOR CLASSIFIED MATERIALS SECURITY COMPLIANCE
 82     YN     C 38639 APPLY OFFICE MANAGEMENT CONCEPTS AND TECHNIQUES
 83     YN     C 38780 MONITOR CORRESPONDENCE AND FORMS TICKLER SYSTEM
 84     YN     C 38786 ADMINISTER CLASSIFIED MATERIAL CONTROL SYSTEM
 85     YN     C 38788 DOCUMENTS THAT REQUIRE OFFICIAL SIGNATURE
 86     YN     C 38861 ORGANIZATION BOOK, BILLS AND PERMANENT DIRECTIVES
 87     YN     C 38862 MANAGEMENT OF ADMINISTRATIVE OFFICE
 88     YN     C 38863 MONITOR ALL REPORTS AND OUTGOING CORRESPONDENCE
 89     YN     C 38864 MONITOR OFFICER FITNESS REPORTS
 90     YN     C 38865 INTERPRET AND ANALYZE MANPOWER AUTHORIZATIONS
 91     YN     C 44462 ESTABLISH TRAINING IN JUMPS AND MAPMIS PROCEDURES
 92     YN     C 65490 ADVISE PAY, ALLOWANCES AND TRAVEL MATTERS
 93     YN     CS35476 MONITOR OPERATIONS IN OWN AREA OF RESPONSIBILITY
 94     YN     CS35478 PREPARE LOCAL DIRECTIVES AND INSTRUCTIONS
 95     YN     CS35480 INSURE MAXIMUM PERSONNEL UTILIZATION
 96     YN     CS35513 SUPERVISE USE OF OFFICE MATERIALS AND EQUIPMENT
 97     YN     CS35514 ADMINISTER RECORDS DISPOSAL PROGRAM
 98     YN     CS35516 MONITOR NAVAL ACTIVITY ADMIN. STAFF FUNCTIONS
 99     YN     CS35517 MONITOR ALL ASPECTS OF RECORDS. FORMS MGMT. PGM.
100     YN     CS35518 ALL ASPECTS OF OFFICER SERVICE RECORDS MAINTENANCE
101     YN     CS44375 TRAINING, EFFECTIVENESS, IMPROVEMENT INITIATION
102     YN     CM35483 CONTROL ADMINISTRATION ACTIVITIES
103     YN     CM35489 FUTURE PLANNING IN OWN AREA OF RESPONSIBILITY
104     YN     CM35490 ESTABLISH GOALS IN OWN AREA OF RESPONSIBILITY
105     YN     CM35492 REVIEW PERSONNEL, EQUIP, MATERIAL REQUIREMENTS
106     YN     CM35519 PERFORM ORGANIZATIONAL ANALYSES
107     YN     CM52298 DEVELOP OPERATING BUDGETS, MONITOR EXPENDITURES
108     BM     3 11511 LOCATE DAMAGE-CONTROL FITTINGS AND EQUIPMENT
109     BM     3 16061 SERVE AS GUN CAPTAIN
110     BM     3 16063 ELEMENTARY FIRE CONTROL
111     BM     3 30051 OPERATION OF DECK WINCHES
112     BM     3 34011 RUNNING RIGGING FOR HANDLING CARGO
113     BM     3 34012 ERECT STATION MARKERS FOR REPLENISHMENT AT SEA
114     BM     3 34013 SERVE AS SIGNALMAN FOR WINCHMAN OR CRANEMAN
115     BM     3 34018 RIGGING AND CARGO-HANDLING EQUIPMENT
116     BM     3 34024 MEMBER OF AN ANCHORING AND MOORING DETAIL
117     BM     3 34031 CHARACTERISTICS OF NAVIGATIONAL LIGHTS AND AIDS
118     BM     3 34036 PREPARE WOOD AND METAL SURFACES FOR PAINTING
119     BM     3 34037 PREPARE AND APPLY PAINTS AND PRIMERS
120     BM     3 34039 USE STENCILS FOR PAINTING LETTERS, NUMERALS
121     BM     3 34047 BASIC KNOT, HITCHES AND SPLICES
122     BM     3 34049 TACKLES AND PURCHASES
123     BM     3 34050 BREAK-OUT AND FAKE-DOWN HAWSERS AND WIRE ROPE
124     BM     3 34052 GROUND TACKLE, RIGGING, HOISTING, STOWING
125     BM     3 34053 HANDLING CANVAS AND LEATHER
126     BM     3 34058 OPERATE VARIOUS TYPES OF LANDING CRAFT
127     BM     3 34067 SERVE AS BOAT COXSWAIN ON LANDING CRAFT
128     BM     3 34068 BOAT LOG, COMPASS, AND CHARTS
129     BM     3 34070 EQUIPMENT ON SHIP'S POWERBOATS, LCVP'S, LCM'S
130     BM     3 34071 BOAT HAILS, RECALLS, SALUTES, AND ETIQUETTE
131     BM     3 34072 CONDUCT MINOR INSPECTION, MAINTENANCE OF BOATS
132     BM     3 34073 MANEUVERING A SMALL BOAT IN A RESTRICTED SPACE
133     BM     3 34410 USE OF NAVIGATIONAL SOUND SIGNALS
134     BM     3 34411 INTERPRET RULES FOR AVOIDING COLLISIONS
135     BM     3 34412 INTERPRET DISTRESS SIGNALS
136     BM     3 34413 GENERAL PRUDENTIAL RULE, GOOD SEAMANSHIP
137     BM     3 34414 RULES TO DETERMINE SAFE WORKING LOADS
```

```
PAGE NUMBER :     3

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDATA

138      BM     3 40001 OPERATE A SEWING MACHINE FOR SEWING CANVAS
139      BM     3 40511 MAINTENANCE OPERATIONS WITH CANVAS
140      BM     3 42024 STAND WATCH AS BOATSWAIN'S MATE OF THE WATCH
141      BM     3 62001 DIRECT HELICOPTERS USING STD. HAND SIGNALS
142      BM     3 86049 INTERPRET SIGNAL FLAGS AND PENNANTS
143      BM     3 86050 SEND AND RECEIVE SEMAPHORE AND FLASHING LIGHT
144      BM     3 94004 STANDING AND RUNNING BLOCKS AND TACKLES
145      BM     3 94592 SHARPEN SCRAPERS AND CHIPPING HAMMERS
146      BM     3 94593 MAINTENANCE OF PAINT-SPRAYING EQUIPMENTS
147      BM     3 94633 LUBRICATE DECK MACHINERY
148      BM     2 11513 DIRECT A DAMAGE-CONTROL REPAIR PARTY SUBGROUP
149      BM     2 16062 SERVE AS MOUNT CAPTAIN
150      BM     2 34014 SUPERVISE LOADING, DISCHARGING, STOWING OF CARGO
151      BM     2 34015 SUPERVISE RIGGING FOR FUELING AT SEA
152      BM     2 34019 RIG YARD-AND-STAY AND SWINGING DERRICKS
153      BM     2 34021 USE OF GROUND TACKLE FOR MOORING AND ANCHORING
154      BM     2 34027 MAINTAIN ANCHORS AND SECURING EQUIPMENT
155      BM     2 34032 RECOGNIZE STORM INDICATIONS
156      BM     2 34033 USE BAROMETER AND WET- AND DRY-BULB THERMOMETERS
157      BM     2 34042 SUPERVISE SHIP'S DECK EQUIPMENT MAINTENANCE
158      BM     2 34043 PAINT REMOVAL, SURFACE PREPARATION, PAINTING
159      BM     2 34044 SELECT PAINTS UNDER DIRECT SUPERVISION
160      BM     2 34051 RIGGING OF ACCOMMODATION LADDER AND BOAT BOOM
161      BM     2 34054 RIG FOR TOWING AND BEING TOWED
162      BM     2 34059 OPERATE BOW RAMP ON LANDING CRAFT
163      BM     2 34060 OPERATE STERN ANCHOR WINCH
164      BM     2 34061 HANDLING OF WIRE DURING BEACHING AND RETRACTING
165      BM     2 34066 INSPECTION, INFLATABLE LIFEBOATS AND EQUIPMENT
166      BM     2 34069 USE OF INFLATABLE LIFEBOAT AND EQUIPMENT
167      BM     2 34415 BOAT HANDLING ON A SHIP IN PORT AND UNDERWAY
168      BM     2 34416 REEVE A SET OF BOAT FALLS
169      BM     2 34417 SUPERVISE SPLICING AND EYE SPLICING
170      BM     2 35326 MAINTAIN PAINTING RECORDS AND REPORTS
171      BM     2 35327 MAINTAIN HULL RECORDS AND REPORTS
172      BM     2 35328 PERSONNEL ASSIGNMENTS FOR PAINTING TASKS
173      BM     2 42342 DIRECT, ASSIGN, INSTRUCT MEN IN DECK WATCH SECTION
174      BM     2 54199 INVENTORY AND SPARE PART SUPPORT IN COSAL
175      BM     2 54200 ORDER REPAIR PARTS USING COSAL
176      BM     2 54201 CHANGE THE ALLOWANCE PARTS LIST
177      BM     2 94005 SUPERVISE GENERAL UPKEEP OF DECK EQUIPMENT
178      BM     1 11514 DIRECT A DAMAGE-CONTROL REPAIR PARTY
179      BM     1 34016 SUPERVISE TRANSFER OF PERSONNEL, CARGO AND MAIL
180      BM     1 34025 DIRECT PROCEDURES FOR MOORING TO A BUOY
181      BM     1 34026 DIRECT PREPARATIONS FOR CARRYING OUT ANCHORS
182      BM     1 34028 PREPARE FOR MAKING MEDITERRANEAN MOOR
183      BM     1 34029 DETERMINE NAVIGATIONAL POSITION
184      BM     1 34030 PLOT DANGER BEARING
185      BM     1 34035 SMALL BOAT HANDLING, EFFECTS OF GETTING ALONGSIDE
186      BM     1 34055 IDENTIFY METHODS OF MINESWEEPING
187      BM     1 34056 CONSTANT TENSION EQUIPMENT RIGGING, UNRIGGING
188      BM     1 34062 IDENTIFY PROCEDURES AND USE OF PONTOON GEAR
189      BM     1 34063 IDENTIFY DUTIES, RESPONSIBILITIES OF BEACHMASTER
190      BM     1 34418 ORGANIZE, SUPERVISE PAINTSHOP AND/OR PAINTLOCKER
191      BM     1 35350 PLAN, ORGANIZE, ADMINISTER DAILY WORK ASSIGNMENTS
192      BM     1 35351 SUPERVISE MAINT OF REQUIRED RECORDS AND LOGS
193      BM     1 44042 TRAIN AND SUPERVISE IN BOATSWAIN'S MATES DUTIES
194      BM     1 54657 POST CHANGES AND ADDITIONS TO COSAL
195      BM     C 11515 ORGANIZE, SUPERVISE A DAMAGE-CONTROL REPAIR PARTY
196      BM     C 16060 SUPERVISE HANDLING OF AMMUNITION AND FLAMMABLES
197      BM     C 34022 SUPERVISE GROUND TACKLE MAINTENANCE
198      BM     C 34023 SUPERVISE ORDINARY MOOR, MOORING WITH ANCHORS
199      BM     C 34057 FORMULATE METHODS OF ASSISTING A STRANDED VESSEL
200      BM     C 34065 IDENTIFY BOATSWAIN'S MATE DUTIES, LCU, WARPING TUG
201      BM     C 34074 USE MANEUVERING BOARD
202      BM     C 34075 USE TIDE AND CURRENT TABLES
203      BM     C 34078 USE NAVIGATIONAL AIDS TO MANEUVER YARD CRAFT
204      BM     C 34079 USE THE THREE-MINUTE RULE
205      BM     C 34081 DIFFERENTIATE INTERNATIONAL, INLAND RULES OF ROAD
206      BM     C 54658 POST SECAS CHANGES AND SUBMIT REPORT TO VFO
```

```
PAGE NUMBER :    4

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDATA

207    BM    CS11516 ASSIST DAMAGE CONTROL OFFICER
208    BM    CS11517 IDENTIFY STABILITY FACTORS
209    BM    CS34076 COORDINATE ALL TYPES OF CARGO
210    BM    CS34077 COORDINATE REPLENISHMENT OF SHIPS AT SEA
211    BM    CS34080 COORDINATE ANCHORING, MOORING DURING STORMS
212    BM    CS35476 TRACK OPERATIONS IN OWN AREA OF RESPONSIBILITY
213    BM    CS35478 PREPARE LOCAL DIRECTIVES AND INSTRUCTIONS
214    BM    CS35479 PREPARE CORRESPONDENCE
215    BM    CS35480 ASSURE MAXIMUM PERSONNEL UTILIZATION
216    BM    CS44375 ORGANIZE AND SCHEDULE TRAINING PROGRAMS
217    BM    CM34082 COORDINATE HANDLING OF BOATS AND LANDING CRAFT
218    BM    CM34083 COORDINATE SALVAGE AND RESCUE OPERATIONS
219    BM    CM35483 PLAN, ORGANIZE, IMPLEMENT AND CONTROL ACTIVITIES
220    BM    CM35489 FORECAST FUTURE PLAN AND ACTION REQUIREMENTS
221    BM    CM35490 ESTABLISH GOALS, OBJECTIVES AND PRIORITIES
222    BM    CM35492 REVIEW PERSONNEL, EQUIP, MATERIAL REQUIREMENTS
223    BM    CM50045 DEPT LONG-RANGE PLANNED MAINTENANCE PROGRAM
224    BM    CM52298 DEVELOP OPERATING BUDGETS, MONITOR EXPENDITURES
225    AN      20350 AIRCRAFT UNDER EMERGENCY CONDITIONS
226    AN      20351 IDENTIFY MARKINGS INDICATING DANGEROUS AREAS
227    AN      20352 IDENTIFY TOXIC PROPERTIES OF CLEANING MATERIALS
228    AN      20368 RECOGNIZE ARMED EJECTION SEATS
229    AN      24425 HANDLE AIRCRAFT BATTERIES
230    AN      25352 KNOW NAVAL AIR ARM GENERAL ORGANIZATION
231    AN      25353 IDENTIFY NAMES, FUNCTIONS OF NAVAL AIR GROUPS
232    AN      25354 IDENTIFY NAVEL AIRCRAFT EQUADRONS
233    AN      25354 IDENTIFY NAVAL AIRCRAFT SQUADRONS
234    AN      40490 FABRICATE TIEDOWN LINES
235    AN      42353 STAND AIRCRAFT SECURITY WATCH
236    AN      46410 USE PUBLICATIONS IN AIRCRAFT MAINTENANCE
237    AN      48330 HOT BRAKES FIRE EQUIPMENT IDENTIFICATION
238    AN      62350 DEFINE COMMON AVIATION TERMS AND NOMENCLATURE
239    AN      62351 IDENTIFY PURPOSE OF BASIC AIRCRAFT INSTRUMENTS
240    AN      62352 USE AIRFRAME STATION NUMBERING SYSTEM
241    AN      62353 DESIGNATION SYSTEM FOR IDENTIFYING NAVAL AIRCRAFT
242    AN      62354 IDENTIFY BASIC TYPES OF AIRCRAFT POWERPLANTS
243    AN      62355 IDENTIFY AC ELECTRONIC, ARMAMENT EQUIPMENTS
244    AN      62357 AC HYDRAULIC, ELECTRICAL, PNEUMATIC, CABLE SYSTEMS
245    AN      62358 IDENTIFY MAJOR STRUCTURAL COMPONENTS OF AIRCRAFT
246    AN      62359 STAND FIRE WATCH DURING FUELING, STARTING
247    AN      62360 COLOR MARKING SYSTEM FOR AIRFIELD VEHICLES
248    AN      62361 DIRECT AIRCRAFT DURING TAXI
249    AN      62362 SERVE AS A MEMBER OF AN AIRCRAFT HANDLING TEAM
250    AN      62363 SECURE AND TIE DOWN AIRCRAFT
251    AN      62364 STAND COCKPIT BRAKE WATCH, TOWING OR PUSHING
252    AN      62365 OPERATING MOBILE EQUIPMENT AROUND AIRCRAFT
253    AN      62366 CARE OF AIRCRAFT HANDLING EQUIPMENT
254    AN      62367 GENERAL METHODS OF HANDLING LAND, CARRIER AIRCRAFT
255    AN      62368 RECOGNIZE AIRCRAFT CARRIER CLOTHING COLOR CODES
256    AN      62369 IDENTIFY, AVIATION SUPPORT EQUIPMENT
257    AN      62370 HANDLE EXTERNAL POWER CABLES FOR AC START, SERVICE
258    AN      62371 OIL, FUEL, HYDRAULICS, NITROGEN/AIR SERVICING
259    AN      62373 SERVICE AIRCRAFT SERVICING EQUIPMENT
260    AN      62376 COLOR CODES USED, AIRCRAFT LINES, TUBING, HOSES
261    AN      79350 CORROSION AND RUST INSPECTION OF AIRCRAFT
262    AN      79356 CLEAN AND WAX AIRCRAFT
263    AN      79357 PLEXIGLASS/FIBERGLASS/RUBBER/FABRIC CLEAN AGENTS
264    AN      94554 USE AND MAINTAIN COMMON HANDTOOLS
265    AN      94555 USE COMMON MEASURING TOOLS
266    AN      94556 INSTALL COMMON AIRCRAFT SAFETY AND LOCKING DEVICES
267    AN      94557 USE COMMON AIRCRAFT SCREWS, NUTS AND BOLTS
268    AN      99350 HANDLING OF AVIATION-TYPE INFLATABLE LIFE VEST
269    AN      99351 ATTACH AND ADJUST PARACHUTE HARNESS
270    AN      99352 OX- EQUIPMENT, LIFE RAFTS
271    AN      99353 IDENTIFY TYPES OF AIRCRAFT SURVIVAL EQUIPMENT
272    SM    3 34009 REEVE AND SPLICE HALYARDS
273    SM    3 42020 RENDER PASSING AND SIDE HONORS
274    SM    3 42021 STAND WATCHES AS A SIGNAL OPERATOR
275    SM    3 50303 USE MAINTENANCE REQUIREMENT CARDS (MRC)
```

```
        PAGE NUMBER :    5

        SOURCE CARD LISTING FOR
        OFFLINE READ  ETAM TDATA
276     SM      3 50928 COMPLETE MAINTENANCE DATA FORMS
277     SM      3 54722 INVENTORY INSTALLED EQUIPMENT
278     SM      3 72001 PERFORM ROUTINE MAINTENANCE ON OPTICAL EQUIPMENT
279     SM      3 77264 SAFEKEEPING AND STORAGE OF CLASSIFIED MATERIAL
280     SM      3 84225 COMMUNICATIONS SECURITY REGULATIONS, PROCEDURES
281     SM      3 86015 RECOGNIZE, USE, INTERPRET EMERGENCY SIGNALS
282     SM      3 86016 TRANSMIT, RECEIVE CODE GROUPS BY FLASHING LIGHT
283     SM      3 86017 TRANSMIT, RECEIVE PLAIN LANGUAGE BY FLASHING LIGHT
284     SM      3 86018 TRANSMIT, RECEIVE PLAIN LANGUAGE BY SEMAPHORE
285     SM      3 86021 DISTINGUISH FLAGHOIST SIGNALS
286     SM      3 86022 SELECT AND DISPLAY FLAGS AND PENNANTS
287     SM      3 86023 EXERCISE RULES OF VISUAL RESPONSIBILITY
288     SM      3 86024 DIRECT DAY LIFEBOAT, NIGHT MAN OVERBOARD EXERCISES
289     SM      3 86025 IDENTIFY FLAGS AND ENSIGNS OF MARITIME NATIONS
290     SM      3 86031 PREPARE A VISUAL MESSAGE FOR TRANSMISSION
291     SM      3 86035 HANDLE MESSAGES ACCORDING TO PRECEDENCE
292     SM      3 86036 PERFORM CHALLENGE REPLY PROCEDURES
293     SM      3 86038 SIGNALMAN DUTIES AS MEMBER OF A BOATCREW
294     SM      3 86365 ELECTRICAL AND ELECTRONIC VISUAL SIGNAL EQUIPMENT
295     SM      3 86366 OPERATE INFRARED AND FLASHING LIGHT EQUIPMENT
296     SM      3 94530 MAINTENANCE OF VISUAL SIGNALING EQUIPMENT
297     SM      2 34007 DRESS AND FULL-DRESS A SHIP
298     SM      2 34008 RECOGNIZE AND USE RULES GOVERNING FLAG DISPLAY
299     SM      2 38654 MAINTAIN VISUAL SIGNAL LOG, VISUAL TRAFFIC FILES
300     SM      2 40469 REPAIR SIGNAL AND PERSONAL FLAGS AND PENNANTS
301     SM      2 42023 LOCATE A SHIP USING FORMATION DIAGRAM
302     SM      2 46036 CORRECT AND MAINTAIN SIGNAL PUBLICATIONS
303     SM      2 50632 COMPLETE PMS FEEDBACK REPORTS
304     SM      2 54200 ORDER COSAL REPAIR PARTS AND SPECIAL TOOLS
305     SM      2 86020 CONSTRUCT AND USE ALL SIGNALING CALL SIGNS
306     SM      2 86033 ENCODE, DECODE NAVAL, INTERNATIONAL SIGNALS
307     SM      2 86044 CLASSIFIED MATERIAL DISPOSAL AND DESTRUCTION
308     SM      2 86045 ENSURE CORRECT CLASSIFIED MATERIAL TRANSPORTATION
309     SM      2 86367 TRANSMIT, RECEIVE FLASHING LIGHT CODE GROUPS
310     SM      2 86368 TRANSMIT, RECEIVE FLASHING LIGHT PLAIN LANGUAGE
311     SM      2 86369 TRANSMIT, RECEIVE SEMAPHORE PLAIN LANGUAGE
312     SM      1 38657 SURVEY AND REQUISITION SIGNAL EQUIPMENT
313     SM      1 38659 PREPARE SIGNALMEN VISUAL COMM. STANDING ORDERS
314     SM      1 44367 PREPARE CRITIQUES OF VISUAL SIGNALING DRILLS
315     SM      1 44450 INSTRUCT IN IDENTIFICATION OF SHIPS AND AIRCRAFT
316     SM      1 44451 INSTRUCT IN VISUAL COMMUNICATIONS PROCEDURES
317     SM      1 44452 INSTRUCT IN RECOGNITION AND AUTHENTICATION
318     SM      1 50986 REVIEW COMPLETED MDCS FORMS
319     SM      1 50987 PREPARE WEEKLY SCHEDULES OF PREVENTIVE MAINTENANCE
320     SM      1 54057 POST CHANGES AND ADDITIONS TO COSAL
321     SM      1 54725 MAINTAIN CUSTODY CARDS AND INVENTORY FILES
322     SM      1 54726 VERIFY SPARE PART SUPPORT IN COSAL
323     SM      1 77266 ADMINISTER SECURITY CLASSIFICATION MANAGEMENT
324     SM      1 86027 AUTHENTICATE RELEASER OF VISUAL MESSAGES
325     SM      1 86028 DIRECT WARTIME TRANSITION, VISUAL SIGNAL METHODS
326     SM      1 86029 CLASSIFY VARIOUS USES OF MESSAGES
327     SM      1 86034 USE RESERVE ON BOARD SIGNALING, PUBLICATIONS
328     SM      1 86370 TRANSMIT, RECEIVE FLASHING LIGHT CODE GROUPS
329     SM      1 86371 TRANSMIT, RECEIVE FLASHING LIGHT PLAIN LANGUAGE
330     SM      1 86372 TRANSMIT, RECEIVE SEMAPHORE PLAIN LANGUAGE
331     SM      C 50988 PREPARE QUARTERLY SCHEDULES, PREVENTIVE MAINT
332     SM      C 54658 POST SECAS CHANGES AND SUBMIT REPORT TO VFO
333     SM      C 77267 ADMINISTER COMMAND SECURITY EDUCATION PROGRAM
334     SM      C 86039 PERFORM AS TACTICAL COMMUNICATIONS WATCH OFFICER
335     SM      C 86373 SUPERVISE VISUAL COMMUNICATIONS ACTIVITIES
336     SM      C 86374 SUPERVISE SIGNAL BRIDGE PERSONNEL
337     SM      CS25371 INTERPRET COMM SECTION ORGANIZATION FUNCTIONS
338     SM      CS35476 MANAGE IN OWN AREA OF RESPONSIBILITY
339     SM      CS35478 PREPARE LOCAL DIRECTIVES AND INSTRUCTIONS
340     SM      CS35479 PREPARE CORRESPONDENCE
341     SM      CS35480 IMPLEMENT MAXIMUM PERSONNEL UTILIZATION
342     SM      CS35496 DISSEMINATE SIGNAL EQUIP TECHNICAL INFORMATION
343     SM      CS35497 OBSERVE, EVALUATE VISUAL COMMUNICATIONS EXERCISES
344     SM      CS35498 ORGANIZE, IMPLEMENT COMMUNICATIONS SECTION
```

```
PAGE NUMBER :    6

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDATA

345     SM     CS44041 ADMINISTER VISUAL COMM EMERGENCY DRILLS
346     SM     CS44375 ORGANIZE AND SCHEDULE TRAINING PROGRAMS
347     SM     CS50989 ADMINISTER LONG-RANGE PLANNED MAINT PROGRAMS
348     SM     CS77268 ADMINISTER COMMAND SECURITY PROGRAM
349     SM     CS86030 CLASSIFY BASEGRAMS AND SPECIAL CATEGORIES
350     SM     CM35483 PLAN, ORGANIZE, IMPLEMENT, CONTROL ACTIVITIES
351     SM     CM35489 FORECAST FUTURE REQUIREMENTS
352     SM     CM35490 ESTABLISH GOAL[illegible]JECTIVES AND PRIORITIES
353     SM     CM35492 REVIEW PERSONN[illegible]EQUIP, MATERIAL REQUIREMENTS
354     SM     CM52298 DEVELOP OPERAT[illegible] BUDGETS, MONITOR EXPENDITURES
355     EN     3 11088 LOCATE PRINCIP[illegible]VALVES, FIRE-MAIN SYSTEM
356     EN     3 11089 OPERATE AND SE[illegible] FIRE AND FLASHING PUMP
357     EN     3 11090 OPERATE INTERN[illegible]COMBUSTION ENGINE-DRIVEN PUMPS
358     EN     3 11091 USE RADIAC INS[illegible]ENTS FOR MONITORING
359     EN     3 11092 USE FIREFIGHTI[illegible]UIPMENT
360     EN     3 15329 LOCATE, REPAIR [illegible]IN HYDRAULICS SYSTEMS
361     EN     3 15330 BLEED HYDRAULIC[illegible]MS
362     EN     3 15331 INSTALL SEALS, [illegible] AND WIPERS IN HYDRAULICS
363     EN     3 18463 USE OF COMMON TEST E[illegible]PMENTS
364     EN     3 24567 OPERATE ELECTRICAL CHA[illegible]ING, REGUL[illegible]ING SYSTEMS
365     EN     3 28085 TECHNICAL DRAWING USAGE ABOARD SHIP
366     EN     3 30317 CONDUCT BOILER FEED WATER TESTS
367     EN     3 30318 CORRECT COMMON BOILER MALFUNCTIONS
368     EN     3 30321 OPERATE STEAM OPERATED DISTILLING PLANTS
369     EN     3 30323 REMOVE SCALE FROM EVAPORATOR TUBES MECHANICALLY
370     EN     3 30328 STAND WATCH ON REFRIGERATION, AIR CONDITIONING
371     EN     3 30336 LOW-, MEDIUM-, HIGH-PRESSURE AIR COMPRESSORS
372     EN     3 30337 ADJUST COMPRESSOR FORCED-FEED LUBRICATORS
373     EN     3 30345 OVERHAUL MANUALLY OPERATED VALVES
374     EN     3 30349 OPERATE AND MAINTAIN HYDRAULIC EQUIPMENT
375     EN     3 30353 TEST ELECTRO-HYDRAULIC STEERING SYSTEM
376     EN     3 30354 PERFORM SHIFTING OF STEERING UNIT PUMPS
377     EN     3 30355 STEERING TRANSFER FROM PILOT HOUSE TO AFT STEERING
378     EN     3 30357 TRANSFER TO MANUAL STEERING
379     EN     3 30361 INSPECT, LUBRICATE AND TEST GALLEY EQUIPMENT
380     EN     3 30363 INSPECT, LUBRICATE AND TEST LAUNDRY EQUIPMENT
381     EN     3 30364 INSPECT, LUBRICATE AND TEST DECK MACHINERY
382     EN     3 30378 MAINTAIN MECHANICAL LUBRICATORS
383     EN     3 30445 REPACK, ADJUST STUFFING BOXES ON PUMPS AND SHAFTS
384     EN     3 31011 COMPLETE OPERATIONAL MAINTENANCE OF DIESEL ENGINES
385     EN     3 31013 OPERATE INTERNAL COMBUSTION ENGINES
386     EN     3 31014 JACKING GEAR ON INTERNAL COMBUSTION ENGINES
387     EN     3 31015 CARRYOUT TURNING OVER OF MAIN ENGINES
388     EN     3 31016 ALIGN LUBRICATING OIL SYSTEM
389     EN     3 31017 USE OF STANDBY LUBRICATING OIL PUMPS
390     EN     3 31018 PURGE DIESEL ENGINE FUEL INJECTION SYSTEM
391     EN     3 31019 ENTRIES TO MAIN PROPULSION/DIESEL GENERATOR LOG
392     EN     3 31021 MAINTAIN ENGINEER'S BELL BOOK
393     EN     3 31024 OPERATION OF LUBRICATION SYSTEM, OPERATING ENGINE
394     EN     3 31026 ALIGN STARTING SYSTEM (AIR/HYDRAULIC/ELECTRICAL)
395     EN     3 31027 ALIGN COOLING SYSTEM
396     EN     3 31028 ALIGN FUEL SYSTEM
397     EN     3 31029 CHEMICAL TESTS, LUBE OIL AND DISTILLATE FUELS
398     EN     3 31030 TESTS ON DIESEL ENGINE CLOSED-COOLING WATER SYSTEM
399     EN     3 31033 OIL AND WATER TESTS, LIGHTING OFF AND SECURING
400     EN     3 31034 ALIGN, OPERATE LUBE OIL AND FUEL OIL PURIFIERS
401     EN     3 31035 CHANGE OIL IN DIESEL ENGINES
402     EN     3 31048 MAINTENANCE OF FLANGE SHIELDS ON PIPING SYSTEMS
403     EN     3 31057 LUBRICATING OIL SYSTEM MAINTENANCE
404     EN     3 31059 SELECT AND USE GRINDING COMPOUNDS
405     EN     3 31072 INTERPRET READINGS OF STANDARD ENGINE INSTRUMENTS
406     EN     3 31079 USE OF STANDARD ENGINEROOM SUPPLIES
407     EN     3 31091 CHECKS, LUBE ON REDUCTION GEARS, THRUST BEARINGS
408     EN     3 31447 LUBE OIL SAMPLES FROM DIESEL ENGINE FOR ANALYSIS
409     EN     3 40529 MAKE UP RIGID TUBING AND LOW PRESSURE HOSES
410     EN     3 46113 USE AND MAINTAIN TECHNICAL AND MAINT MANUALS
411     EN     3 51011 COMPLETE MAINTENANCE DATA FORMS
412     EN     3 51026 USE MAINTENANCE REQUIREMENT CARDS (MRC)
413     EN     3 54800 IDENTIFY CATEGORIES OF MATERIAL IN COSAL
```

```
PAGE NUMBER :    7

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDATA

414    EN    3 94155 USE AND CARE FOR MEASURING INSTRUMENTS
415    EN    3 98024 ADJUST FUEL COMBUSTION TO MINIMIZE AIR POLLUTION
416    EN    3 98025 DETERMINE POLLUTION THREATS DUE TO PETROLEUM SPILL
417    EN    3 98026 PREVENTION OF OIL SPILL ACCIDENTS
418    EN    3 98335 SAFETY IN HIGH NOISE AND HEAT STRESS AREAS
419    EN    2 11093 INSPECT AND REPAIR PORTABLE PUMPS
420    EN    2 11447 IDENTIFY ORGANIZATION OF A REPAIR PARTY
421    EN    2 30319 OPERATE AUXILIARY BOILER
422    EN    2 30324 REMOVE SCALE FROM DISTILLING UNITS CHEMICALLY
423    EN    2 30325 TEST EVAPORATORS AND CONDENSERS FOR LEAKS
424    EN    2 30326 CONDUCT STANDARD DISTILLING PLANT TESTS
425    EN    2 30330 HALIDE TORCH TESTS ON REFRIGERATION UNITS
426    EN    2 30331 OIL CHANGES IN REFRIGERATION COMPRESSORS
427    EN    2 30332 SUCTION, DISCHARGE VALVE TESTS, REFRIG COMPRESSORS
428    EN    2 30333 TEST AND RECHARGE REFRIGERATION UNITS
429    EN    2 30334 TEST, RENEW OIL SEALS IN REFRIGERATION UNITS
430    EN    2 30338 SUCTION, DISCHARGE VALVES MAINT ON AIR COMPRESSORS
431    EN    2 30341 DETERMINE CLEARANCES IN PUMPS
432    EN    2 30342 OVERHAUL PUMPS
433    EN    2 30343 CHECK FOR ALIGNMENT OF PUMP DRIVING UNITS
434    EN    2 30344 REPLACE ROTARY SEALS
435    EN    2 30347 TEST HYDROSTATICALLY, REPAIR PIPING SYSTEMS
436    EN    2 30358 LUBRICATE STEERING GEAR
437    EN    2 30372 ADJUST, REPAIR REFRIGERATION AUTOMATIC CONTROLS
438    EN    2 30374 CORRECT INEFFICIENT REFRIGERATION SYSTEMS OPS
439    EN    2 30911 CLEAN, INSPECT, TEST HEAT EXCHANGERS
440    EN    2 30912 MAINT OF EXHAUST SILENCERS AND SPARK ARRESTERS
441    EN    2 30913 OPERATE A HEAT RECOVERY DISTILLING PLANT
442    EN    2 31038 INTERNAL COMBUSTION ENGINE MAINTENANCE
443    EN    2 31039 REFACE, RESEAT INTAKE AND EXHAUST VALVES
444    EN    2 31043 TEST UNIT INJECTORS AND/OR FUEL INJECTION NOZZLES
445    EN    2 31044 MAINTENANCE OF FUEL OIL INJECTORS
446    EN    2 31049 PREVENTION OF CRANKCASE EXPLOSIONS
447    EN    2 31051 TROUBLESHOOT GASOLINE ENGINES
448    EN    2 31061 CHEMICAL ANALYSIS OF FUELS AND OILS
449    EN    2 31092 ENGINE AND DRIVE SHAFT ADJUSTMENTS, SMALL CRAFT
450    EN    2 31093 CHECK OIL CLEARANCE IN BEARINGS
451    EN    2 31448 MAINT INTERNAL COMBUSTION ENGINE TURBO-CHARGERS
452    EN    2 31449 TROUBLESHOOT INTERNAL COMBUSTION ENGINE BLOWERS
453    EN    2 31450 OPERATE AND MAINTAIN VERTICAL CONVEYORS
454    EN    2 31451 MAINT OF LUBE OIL AND FUEL OIL PURIFIERS
455    EN    2 31452 EXTERNAL ADJUSTMENTS ON DIESEL ENGINES
456    EN    2 31453 MAINT TREND ANALYSIS DATA ON DIESEL ENGINE OPS
457    EN    2 35112 RECORD RESULTS OF EQUIPMENT TESTS
458    EN    2 35342 RECORD TRAINING DATA
459    EN    2 48245 TROUBLESHOOT AUXILIARY SYSTEMS
460    EN    2 48256 DYE-PENETRANT AND MAGNAFLUX TESTS, METAL SURFACES
461    EN    2 51114 COMPLETE PMS FEEDBACK REPORTS
462    EN    2 54801 INSTALLED EQUIP AND SPARE PART SUPPORT IN COSAL
463    EN    2 54802 ORDER REPAIR PARTS AND SPECIAL TOOLS USING COSAL
464    EN    2 94156 USE MECHANICAL MAINTENANCE TEST EQUIPMENT
465    EN    2 94157 PLAIN TURNING AND CUTTING ON AN ENGINE LATHE
466    EN    1 11454 SUPERVISE A DAMAGE CONTROL PARTY
467    EN    1 30320 MAINTENANCE AND TESTS ON AUXILIARY BOILERS
468    EN    1 30327 PLUG AND/OR REPLACE HEAT EXCHANGER TUBES
469    EN    1 30339 INTERCOOLER, AFTERCOOLER, OIL COOLER MAINTENANCE
470    EN    1 30340 OVERHAUL OF AIR COMPRESSORS
471    EN    1 30348 REDUCING, RELIEF, TEMPERATURE CONTROL VALVE MAINT
472    EN    1 30350 HYDRAULIC SYSTEM MAINTENANCE
473    EN    1 30914 STEAM OPERATED DISTILLING PLANT MAINTENANCE
474    EN    1 30915 VAPOR COMPRESSION DISTILLING PLANT MAINTENANCE
475    EN    1 31012 INTERNAL COMBUSTION ENGINE MAINTENANCE
476    EN    1 31025 ADJUSTMENT OF TEMPERATURE REGULATING VALVES
477    EN    1 31031 GRAPHICALLY DETERMINE PROPER ENGINE OPERATION
478    EN    1 31037 REPAIR OF INTERNAL COMBUSTION ENGINES
479    EN    1 31040 CHECK MAIN ENGINE BEARING AND THRUST CLEARANCES
480    EN    1 31041 BLOWER CLEARANCES, LOBES, TIMING, DRIVE GEAR CHECK
481    EN    1 31042 INSPECT, TAKE CLEARANCES ON TURBO-CHARGERS
482    EN    1 31094 CHECK THRUST BEARING READINGS
```

```
PAGE NUMBER :    8

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDATA

483    EN     1 31097 CHECK DAMAGE TO SHAFTS AND THRUST BEARINGS
484    EN     1 31253 INSPECT, ADJUST GOVERNORS AND OVERSPEED TRIPS
485    EN     1 31454 COMPLETE MAINT OF GOVERNORS AND OVERSPEED TRIPS
486    EN     1 35323 MAINTAIN ENGINEROOM RECORDS
487    EN     1 51167 REVIEW COMPLETED MDCS FORMS
488    EN     1 51168 PREPARE WEEKLY SCHEDULES OF PREVENTIVE MAINTENANCE
489    EN     1 54827 POST CHANGES AND ADDITIONS TO COSAL
490    EN     C 30359 MECHANICAL OPERATION TESTS
491    EN     C 30674 SUPERVISE AND TRAIN IN MECHANICAL MAINTENANCE
492    EN     C 31045 DIESEL AND GASOLINE ENGINES MAINTENANCE
493    EN     C 31046 CONDUCT DIESEL ENGINE POST-OVERHAUL CHECKOUT
494    EN     C 31047 MEASURE CRANKSHAFT DEFLECTION
495    EN     C 31077 SHIP'S EXTERNAL DRIVE CHECKS WHEN IN DRYDOCK
496    EN     C 31095 MAIN REDUCTION GEAR BACKLASH, ALIGNMENT CHECKS
497    EN     C 35432 PREPARE POST REPAIR TRIALS REPORTS
498    EN     C 35433 REVIEW ALL ENGINE ROOM RECORDS
499    EN     C 35885 SUPERVISE ALL ADMINISTRATION FOR ENGINEERING DEPT
500    EN     C 35886 INSPECT RESULTS OF EQUIPMENT TESTS
501    EN     C 42278 INTERPRET DUTIES, ENGINEERING OFFICER OF THE WATCH
502    EN     C 42279 SUPERVISE ENGINEROOM WATCH ON DIESEL-DRIVEN SHIP
503    EN     C 42367 INTERPRET DUTIES, ENGINEERING OFFICER OF THE WATCH
504    EN     C 51069 ESTIMATE REPAIRS FOR AUX, MAIN PROPULSION  MACH
505    EN     C 51169 PREPARE QUARTERLY PM SCHEDULES
506    EN     C 51475 INSPECT "SHIP TO SHOP" WORK AND REPORTS
507    EN     C 51476 ANALYZE CHEMICAL OIL ANALYSIS REPORTS
508    EN     C 51477 INTERPRET DIESEL ENGINE TREND ANALYSIS DATA
509    EN     C 51478 INTERPRET DIESEL OIL CHEMICAL TESTS
510    EN     C 54804 ENTER CHANGES TO SECAS, SUBMIT REPORTS TO VFO
511    EN     C 98021 SUPERVISE OIL SPILL CONTAINMENT EXERCISES
512    EN     CS31410 TRAINING OF PERSONNEL IN MECHANICAL MAINTENANCE
513    EN     CS31455 ANALYSIS OF DAILY OPERATING RECORDS
514    EN     CS31456 ESTIMATE FEASIBILITY OF REPAIRS
515    EN     CS35072 MONITOR MAINT OF EQUIPMENT, MATERIAL LOGS
516    EN     CS35362 ADVISE PERSONNEL IN REPAIR PAPERWORK
517    EN     CS35476 MANAGEMENT IN OWN AREA OF RESPONSIBILITY
518    EN     CS35478 PREPARE LOCAL DIRECTIVES AND INSTRUCTIONS
519    EN     CS35479 PREPARE CORRESPONDENCE
520    EN     CS35480 ESTABLISH MAXIMUM PERSONNEL UTILIZATION
521    EN     CS35884 REVIEW WATCHSTANDING QUALIFICATIONS
522    EN     CS35887 REVIEW PUBLICATIONS REQUIREMENTS
523    EN     CS44375 ORGANIZE AND SCHEDULE TRAINING PROGRAMS
524    EN     CS51073 ORGANIZE AND SCHEDULE WORK
525    EN     CS51206 ADMINISTER LONG-RANGE PLANNED MAINTENANCE PROGRAM
526    EN     CS98280 MONITOR ENVIRONMENTAL POLLUTION CONTROL PROGRAMS
527    EN     CS98281 MONITOR ENVIRONMENTAL PROGRAMS
528    EN     CM20060 DEVELOP AND MONITOR SAFETY PROGRAMS
529    EN     CM31457 MONITOR DAILY OPERATING RECORDS
530    EN     CM35365 PREPARE MONTHLY SUMMARY, DIESEL-DRIVEN SHIPS
531    EN     CM35366 IC AND GAS TURBINE ENGINE PUBLICATION DISTRIBUTION
532    EN     CM35483 PLAN, ORGANIZE, IMPLEMENT, CONTROL ACTIVITIES
533    EN     CM35489 FORECAST FUTURE REQUIREMENTS
534    EN     CM35490 ESTABLISH GOALS, OBJECTIVES AND PRIORITIES
535    EN     CM35492 REVIEW PERSONNEL, EQUIPMENT, MATERIAL REQUIREMENTS
536    EN     CM46616 REVIEW MANUAL, PRINT, PUBLICATION REQUIREMENTS
537    EN     CM51072 MONITOR ALL "SHIP TO SHOP" WORK AND REPORTS
538    EN     CM51075 PERFORM CASUALTY ANALYSIS
539    EN     CM51179 COORDINATE REPAIRS BETWEEN SHIP AND SHIPYARD
540    EN     CM52298 DEVELOP OPERATING BUDGETS, MONITOR EXPENDITURES
541    EN     CM54806 MONITOR ASSIGNED EQUIPMENT LOGISTICS SUPPORT
542    GMG    3 14086 TEST ELECTRONIC TUBES
543    GMG    3 14564 INTERPRET COLOR-CODING OF ELECTRONIC PARTS
544    GMG    3 14565 TEST AND REPLACE ELECTRONIC PARTS
545    GMG    3 14570 IDENTIFY ELECTRONIC CIRCUIT PROTECTIVE DEVICES
546    GMG    3 15305 INSPECT AND SERVICE HYDRAULIC EQUIPMENT
547    GMG    3 15307 TEST HYDRAULIC ORDNANCE EQUIPMENT
548    GMG    3 15314 RECOGNIZE TYPES AND PRINCIPLES, PRESSURE GAUGES
549    GMG    3 15315 MAINTAIN PIPES, FITTINGS, SEALS AND GASKETS
550    GMG    3 16027 IDENTIFY TYPES OF EXPLOSIVE ORDNANCE
551    GMG    3 16033 HANDLE, STOW ORDNANCE AND PYROTECHNICS
```

```
PAGE NUMBER :    9

SOURCE CARD LISTING FOR
OFFLINE READ  FTAM TDATA
552     GMG   3 16034 STOW, ISSUE, INVENTORY SMALL ARMS AND AMMUNITION
553     GMG   3 16304 OPERATE AMMUNITION HOIST UNITS
554     GMG   3 16334 SERVICE ORDNANCE HANDLING AND ASSOCIATED EQUIPMENT
555     GMG   3 16398 INSTALL, SET, REMOVE PROJECTILE FUZING DEVICES
556     GMG   3 16399 ASSEMBLE AND USE LANDING PARTY EQUIPMENT
557     GMG   3 16400 OPERATE AND MAINTAIN LINE-THROWING GUNS
558     GMG   3 16402 PERFORM FUZING OF HAND GRENADES
559     GMG   3 16403 MONITOR, REPLACE RELATIVE HUMIDITY INDICATORS
560     GMG   3 16404 LOG MAGAZINE TEMPERATURES, INSPECT MAGAZINE AREA
561     GMG   3 16405 OPERATE MAGAZINE FIREFIGHTING SYSTEMS
562     GMG   3 16407 INSPECT ORDNANCE ITEMS
563     GMG   3 16408 OPERATE PYROTECHNIC EQUIPMENT
564     GMG   3 17079 MAINTAIN SMALL ARMS
565     GMG   3 17081 MAINTAIN LANDING PARTY EQUIPMENT
566     GMG   3 17365 PERFORM DAILY TRANSMISSION TESTS
567     GMG   3 18017 OPERATE MECHANICAL AND ELECTRICAL TEST EQUIPMENT
568     GMG   3 18022 OPERATE PRESSURE GAUGES, INTERPRET READINGS
569     GMG   3 24023 MAKE ELECTRICAL CABLE AND WIRING CHECKS
570     GMG   3 24025 COMMUTATOR, SLIPPING, BRUSH ASSEMBLY MAINTENANCE
571     GMG   3 24390 ELECTRICAL/ELECTRONIC TERMS/UNITS OF MEASURE
572     GMG   3 24391 IDENTIFY TYPES OF A.C. MOTORS
573     GMG   3 24497 IDENTIFY TYPES OF D.C. MOTORS AND GENERATORS
574     GMG   3 24500 CLEAN ELECTRICAL CONTACTS AND SWITCHES
575     GMG   3 24501 MECHANICAL WIRE CONNECTING, INCLUDING SOLDERING
576     GMG   3 24502 DETECT AND REPLACE BLOWN FUSES
577     GMG   3 24503 POINT-TO-POINT VOLTAGE, RESISTANCE MEASUREMENTS
578     GMG   3 28016 IDENTIFY MECHANICAL, HYDRAULIC, ELECTRONIC SYMBOLS
579     GMG   3 28019 USE AND READ DIAGRAMS AND DRAWINGS
580     GMG   3 28020 TRACE CIRCUITS ON SCHEMATICS AND DRAWINGS
581     GMG   3 29236 MAINTAIN ONBOARD SYSTEMS BREECH MECHANISMS
582     GMG   3 29238 REPAIR SEALING SURFACES, MATING AREAS, THREADS
583     GMG   3 29239 REPLACE FAULTY FASTENING HARDWARE AND SEALS
584     GMG   3 29549 INSTALL ALIGNMENT EQUIPMENT ON GUN MOUNTS
585     GMG   3 40525 CUT, BEND, PREPARE PIPING AND TUBING
586     GMG   3 46030 USE PUBS RELATED TO OPS/TEST/REPAIR, ORDNANCE EQPT
587     GMG   3 48241 HANDLING AND DISPOSAL OF MISFIRED ROUNDS
588     GMG   3 51011 COMPLETE MAINTENANCE DATA FORMS
589     GMG   3 51026 USE MAINTENANCE REQUIREMENT CARDS
590     GMG   3 54192 PROCESS AND DOCUMENT REPAIRABLE ITEMS
591     GMG   3 54759 MAINTAIN CONSUMABLE SUPPLIES, OBTAIN REPLACEMENTS
592     GMG   3 54760 IDENTIFY MATERIALS CONTAINED IN COSAL
593     GMG   3 54761 INVENTORY TOOLS AND PORTABLE TESTING EQUIPMENT
594     GMG   3 79272 CORROSION PREVENTION, MOISTURE PROTECTIVE MATERIAL
595     GMG   3 92336 USE, MAINTAIN FIXED/PORTABLE PNEU/ELEC POWER TOOLS
596     GMG   3 94424 USE, MAINTAIN HANDTOOLS
597     GMG   3 94508 OPERATE TORQUING TOOLS
598     GMG   2 14043 SERVOAMPLIFIERS MAINTENANCE
599     GMG   2 14566 ANALYZE ELECTRONIC SYSTEMS MALFUNCTIONS
600     GMG   2 14567 FIRE CONTROL INTERFACE DSOT
601     GMG   2 14798 CHECK ALARMS, SENSING DEVICES FOR OPERATION
602     GMG   2 15023 PERFORM HYDRAULIC TESTS, ORDNANCE EQUIPMENT
603     GMG   2 15309 INSTALL, MAINTAIN COMPONENTS IN HYDRAULIC SYSTEMS
604     GMG   2 15311 ANALYZE PUMP MALFUNCTIONS AND MAKE REPAIRS
605     GMG   2 15312 RECOIL, COUNTER RECOIL, GAS EJECTOR SYSTEMS
606     GMG   2 16004 USE DEMOLITION CHARGES FOR EMERGENCY DESTRUCTION
607     GMG   2 16021 INSPECT EXPLOSIVE ORDNANCE ITEMS FOR DAMAGE
608     GMG   2 16022 INSTRUCT CREWS IN HANDLING AMMUNITION, EXPLOSIVES
609     GMG   2 16054 HANDLE DAMAGED EXPLOSIVE COMPONENTS
610     GMG   2 16099 CHECK ALARMS AND SENSING DEVICES
611     GMG   2 17366 WEAPONS ADJUSTMENT AND ALIGNMENT
612     GMG   2 17367 REPLACE DEFECTIVE COMPONENTS OR MODULES
613     GMG   2 18419 INSTALL, OPERATE GUN MOUNT DYNAMIC TEST EQUIPMENT
614     GMG   2 24026 TEST, REPLACE ELECTRICAL CIRCUIT COMPONENTS
615     GMG   2 24027 REMOVE, INSTALL, ALIGN, ZERO SYNCHROS
616     GMG   2 24393 ANALYZE ORDNANCE ELECTRICAL SYSTEMS MALFUNCTIONS
617     GMG   2 28017 PREPARE BASIC DRAWINGS AND MATHEMATICAL LAYOUTS
618     GMG   2 28025 USE WORKING DRAWINGS IN MAKING REPAIRS, MODS
619     GMG   2 28382 USE MICROFICHE, LD DRAWINGS, READ/PRINT MACHINES
620     GMG   2 29237 CHECK ACCURACY OF TRAIN AND ELEVATION INDICATORS
```

```
PAGE NUMBER :  10

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDATA

621     GMG  2 29243 TEST AND MAINTAIN ORDNANCE WATER SYSTEM
622     GMG  2 29245 PREPARE ONBOARD GUNS AND EQUIPMENT FOR FIRING
623     GMG  2 29246 PERFORM POST FIRING PROCEDURES ON GUNS, EQUIPMENT
624     GMG  2 29248 TEST FIRING CUTOJT ASSEMBLIES, INTERPRET MEANINGS
625     GMG  2 29249 TEST FUZE-SETTING DEVICES, INTERPRET MEANINGS
626     GMG  2 29250 INSPECT, TEST ASSEMBLIES (LESS POWER DRIVES)
627     GMG  2 29251 MAINTAIN, REPAIR ASSEMBLIES (LESS POWER DRIVES)
628     GMG  2 29252 MAINTAIN, CHECK JRDNANCE OPERATION LOGS
629     GMG  2 36450 MAINTAIN SHOP AND EQUIPMENT WORKLOGS
630     GMG  2 36451 MAINTAIN EQUIPMENT HISTORIES
631     GMG  2 51114 COMPLETE PMS FEEDBACK REPORTS
632     GMG  2 54708 SUBMIT DAILY AMMUNITION TRANSACTION REPORTS
633     GMG  2 54813 CORRECT THE COSAL REPORTS
634     GMG  2 54814 INVENTORY EQUIPMENT, SPARE PART SUPPORT IN COSAL
635     GMG  2 54815 ORDER PARTS AND TOOLS USING COSAL
636     GMG  1 14062 DIAGNOSE ELECTRONIC MALFUNCTIONS
637     GMG  1 15313 TEST, ADJUST, REPAIR HYDRAULIC MECHANISMS
638     GMG  1 16023 SUPERVISE CREWS IN ORDNANCE HANDLING
639     GMG  1 18018 ANALYZE, INTERPRET DATA, GUN MOUNT DYNAMIC TEST
640     GMG  1 18021 OPERATE AN OSCILLOSCOPE
641     GMG  1 28018 PREPARE ADVANCED DRAWINGS AND MATHEMATICAL LAYOUTS
642     GMG  1 29253 INSPECT AND TEST POWER DRIVES
643     GMG  1 29254 MAINTAIN AND REPAIR POWER DRIVES
644     GMG  1 29255 REPAIR AND ADJUST FIRING CUTOUT ASSEMBLIES
645     GMG  1 29256 REPAIR AND ADJUST FUZE-SETTING DEVICES
646     GMG  1 29551 TEST, ADJUST, REPAIR ALARMS AND SENSING DEVICES
647     GMG  1 29552 OVERHAUL AND REPAIR GUN MOUNTS
648     GMG  1 38344 COMPLETE WEAPONS SYSTEM DEFICIENCY REPORTS
649     GMG  1 38345 PREPARE PERIODIC PERFORMANCE REPORTS
650     GMG  1 38346 MAINTAIN AMMUNITION RECORDS, INVENTORIES
651     GMG  1 38887 MAINTAIN RECORDS, CONVENTIONAL EXPLOSIVE ORDNANCE
652     GMG  1 38888 ORGANIZE, SUPERVISE MAINTENANCE OF A TECHNICAL LIBRARY
653     GMG  1 511[illegible]7 REVIEW COMPLETED MDCS FORMS
654     GMG  1 511[illegible] PREPARE WEEKLY SCHEDULES OF PREVENTIVE MAINTENANCE
655     GMG  1 54816 EFFECT CHANGES TO SECAS SYSTEM
656     GMG  1 54843 INITIATE SURVEYS AND DAMAGED SHIPMENT REPORTS
657     GMG  1 54844 SUPERVISE INVENTJRIES AND MAINTAIN CUSTODY RECORDS
658     GMG  1 98255 PREPARE ENVIRONMENTAL POLLUTION CONTROL REPORTS
659     GMG  C 14042 REPAIRS TO INDICATOR AND RECEIVER REGULATORS
660     GMG  C 14044 SUPERVISE MAINT OF ORDNANCE ELECTRIC/ELECTRONIC SYS
661     GMG  C 14045 INSPECT REPAIRS, ELECTRIC/ELECTRONIC EQUIP
662     GMG  C 14064 ANALYZE ELECTRIC/ELECTRONIC DISCREPANCY TRENDS
663     GMG  C 16032 SUPERVISE ASHORE STOWAGE OF EXPLOSIVE ORDNANCE
664     GMG  C 16411 SUPERVISE AND INSPECT ORDNANCE STOWAGE
665     GMG  C 17426 DIAGNOSE EFFECTS OF ENVIRONMENTAL CONDITIONS
666     GMG  C 17427 INSPECT COMPLETED MODERNIZATIONS
667     GMG  C 20276 SUPERVISE SAFETY PROCEDURES FOR INSPECTIONS
668     GMG  C 38892 PREPARE WEAPONS DEPARTMENT REPORTS
669     GMG  C 38893 IMPLEMENT LOADING AND STOWAGE PLANS
670     GMG  C 38897 INTERPRET REQUIREMENTS FOR NAVY INSPECTIONS
671     GMG  C 51079 SUPERVISE WORK AND ESTABLISH WORK PRIORITIES
672     GMG  C 51169 PREPARE PREVENTIVE MAINTENANCE QUARTERLY SCHEDULES
673     GMG  C 54766 PERFORM ARMAMENT INVENTORIES
674     GMG  C 98336 SUPERVISE SHIPBOARD ENVIRONMENTAL POLLUTION CONT
675     GM   CS17417 SUPERVISE ALIGNMENT OF GUN, MISSILE BATTERIES
676     GM   CS35341 PREPARE, SUBMIT ALL ADMINISTRATIVE REPORTS
677     GM   CS35476 REPORT OPERATIONS IN OWN AREA OF RESPONSIBILITY
678     GM   CS35478 PREPARE LOCAL DIRECTIVES AND INSTRUCTIONS
679     GM   CS35479 PREPARE CORRESPONDENCE
680     GM   CS35480 ESTABLISH MAXIMUM PERSONNEL UTILIZATION
681     GM   CS44375 ORGANIZE AND SCHEDULE TRAINING PROGRAMS
682     GM   CS51153 ESTIMATE TIME, MATERIAL, LABOR REQUIREMENTS
683     GM   CS51154 SUPERVISE INSPECTION PROCEDURES
684     GM   CS51155 REVIEW, EVALUATE TEST/CHECKOUT PROCEDURES
685     GM   CS51206 ADMINISTER A LONG-RANGE PM PROGRAM
686     GM   CS54196 CONTROL ORDNANCE PROCUREMENT EXPENDITURES
687     GM   CM20291 COJRDINATE, MANAGE DEPARTMENT ORDNANCE SAFETY
688     GM   CM35483 PLAN, ORGANIZE, IMPLEMENT, CONTROL ACTIVITIES
689     GM   CM35489 FORECAST FUTURE REQUIREMENTS
```

```
PAGE NUMBER :  11

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDATA
690     GM     CM35490 ESTABLISH GOALS, OBJECTIVES AND PRIORITIES
691     GM     CM35492 REVIEW PERSONNEL, EQUIPMENT, MATERIAL REQMNTS
692     GM     CM35551 SUPERVISE OPERATIONS AND PROCEDURES
693     GM     CM35552 PREPARE MAINT PERIODIC AND RECURRING REPORTS
694     GM     CM35553 DEVELOP ACCOUNTING PROCEDURES FOR ALL WORK
695     GM     CM51156 COORDINATE SURFACE WEAPONS SYSTEMS OVERHAUL
696     GM     CM51157 ANALYZE DISCREPANCY TRENDS IN EQUIPMENT RECORDS
697     GM     CM52298 DEVELOP OPERATING BUDGETS, MONITOR EXPENDITURES
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDSCR


 1     OFFLINE READ  ETAM TDSCR
 2     SM    3 34009 41 59
 3     SM    3 42020 41 42 62
 4     SM    3 42021 41 58
 5     SM    3 50303 01
 6     SM    3 50928 01
 7     SM    3 54722 01
 8     SM    3 72001 41 52 59 61
 9     SM    3 77264 02
10     SM    3 84225 53 55 60 61
11     SM    3 86015 53 55 60 61
12     SM    3 86016 55 60 61
13     SM    3 86017 55 60 61
14     SM    3 86018 55 60 61
15     SM    3 86021 55 61
16     SM    3 86022 41 60
17     SM    3 86023 41 54
18     SM    3 86024 42 60 62 91
19     SM    3 86025 53
20     SM    3 86031 41 60 82
21     SM    3 86035 41 60
22     SM    3 86036 41 60
23     SM    3 86038 43 60
24     SM    3 86365 52 54 61
25     SM    3 86366 41 42 57 61
26     SM    3 94530 41 59 61
27     YN    3 38605 02
28     YN    3 38606 01
29     YN    3 38822 05
30     YN    3 38823 01 02 04 26
31     YN    3 38824 35
32     YN    3 38825 23
33     YN    3 38826 21
34     YN    3 38827 03
35     YN    3 38830 01 12
36     YN    3 38831 21
37     YN    3 38834 11
38     YN    3 38835 01
39     YN    3 38843 02
40     YN    3 38852 11 21
41     YN    3 38958 01
42     YN    3 46358 02 12
43     YN    3 77269 41 42 61
44     YN    3 77284 03
45     YN    3 94620 21 23 41 59
46     YN    2 25269 53 55
47     YN    2 38619 02
48     YN    2 38620 02 04
49     YN    2 38627 02 04
50     YN    2 38836 01 04 11 21
51     YN    2 38840 21
52     YN    2 38841 32
53     YN    2 38842 02
54     YN    2 38844 02 04
55     YN    2 38846 12
56     YN    2 38848 53
57     YN    2 38849 11
58     YN    2 38850 11
59     YN    2 38851 01
60     YN    2 38853 04
61     YN    2 38854 02 04
62     YN    2 38859 01
63     YN    2 38957 02
64     YN    2 38959 01 11
65     YN    2 46360 01
66     YN    2 65015 12 13 35
67     YN    2 65500 12 35
68     YN    2 65501 12 35
```

```
         PAGE NUMBER :    2

         SOURCE CARD LISTING FOR
         OFFLINE READ  ETAM TDSCR

  69    YN    1 38629 41 42 58 62 82
  70    YN    1 38630 01 02 33
  71    YN    1 38631 01
  72    YN    1 38634 01 12
  73    YN    1 38855 11 13
  74    YN    1 38856 21
  75    YN    1 38858 01 11 35
  76    YN    1 38860 11
  77    YN    1 38960 11
  78    YN    1 38961 11
  79    YN    1 46361 02 04
  80    YN    1 65480 35
  81    YN    1 77285 02
  82    YN    C 38639 12 14 33
  83    YN    C 38780 02 33
  84    YN    C 38786 04 33
  85    YN    C 38788 04
  86    YN    C 38861 02 11 12 33
  87    YN    C 38862 14 33
  88    YN    C 38863 12
  89    YN    C 38864 02 12
  90    YN    C 38865 12
  91    YN    C 44462 13 14 32 33
  92    YN    C 65490 35
  93    YNCS    35476 12 35
  94    YNCS    35478 12 13 14
  95    YNCS    35480 13 14 33 34
  96    YNCS    35513 33
  97    YNCS    35514 33
  98    YNCS    35516 33
  99    YNCS    35517 12
 100    YNCS    35518 12
 101    YNCS    44375 13 14 32
 102    YNCM    35483 12 14 32 33 34
 103    YNCM    35489 12 14
 104    YNCM    35490 12 13 14
 105    YNCM    35490 12
 106    YNCM    35519 12
 107    YNCM    52298 14
 108    BM    3 11511 54
 109    BM    3 16061 33 61
 110    BM    3 16063 41 42 57 58 61
 111    BM    3 30051 57 61 91
 112    BM    3 34011 42 57 61
 113    BM    3 34012 41 57 58
 114    BM    3 34013 43 91
 115    BM    3 34018 41 42 58 59 72
 116    BM    3 34024 43 57 61
 117    BM    3 34031 53 54
 118    BM    3 34036 41 52 59 61
 119    BM    3 34037 41 57 59
 120    BM    3 34039 41 57 59
 121    BM    3 34047 41 57 59 61 81
 122    BM    3 34049 59 61
 123    BM    3 34050 41 59 61
 124    BM    3 34052 54 60
 125    BM    3 34053 41 54 56 58
 126    BM    3 34058 42 57 61 91
 127    BM    3 34067 33 42 57 60 61 91
 128    BM    3 34068 41 53 55 56 58 60
 129    BM    3 34070 53 54
 130    BM    3 34071 41 60
 131    BM    3 34072 41 52 81
 132    BM    3 34073 42 56 57 61
 133    BM    3 34410 54 56
 134    BM    3 34411 41 55
 135    BM    3 34412 55
 136    BM    3 34413 55 58
 137    BM    3 34414 42 56 61
```

```
              PAGE NUMBER :    3

              SOURCE CARD LISTING FOR
              OFFLINE READ  ETAM TDSCR

   138     BM     3 40001 57 61 81
   139     BM     3 40511 57 61 81
   140     BM     3 42024 41 52 54 56 60 72
   141     BM     3 62001 42 60 62
   142     BM     3 86049 53 55
   143     BM     3 86050 60 61
   144     BM     3 94004 57 59 61 81
   145     BM     3 94592 57 61
   146     BM     3 94593 41 57 61
   147     BM     3 94633 41 59 61
   148     BM     2 11513 33 42 58 61 71 72
   149     BM     2 16062 33 41 58 61
   150     BM     2 34014 33 42 61 72
   151     BM     2 34015 33 42 58 61 72
   152     BM     2 34019 41 58 59
   153     BM     2 34021 33 41 58
   154     BM     2 34027 41 52 54 57 59 61
   155     BM     2 34032 54 56
   156     BM     2 34033 54 58
   157     BM     2 34042 33 41 57 61
   158     BM     2 34043 32 33 41 59 61
   159     BM     2 34044 72
   160     BM     2 34051 33 41 59 61
   161     BM     2 34054 58 59 91
   162     BM     2 34059 41 57 61
   163     BM     2 34060 42 57 91
   164     BM     2 34061 52 54 58 60
   165     BM     2 34066 33 52 54 61 81
   166     BM     2 34069 33 41 57 59 61
   167     BM     2 34415 33 41 57 61
   168     BM     2 34416 57 59 61
   169     BM     2 34417 33 57 59 61 81
   170     BM     2 35326 01 02
   171     BM     2 35327 01 02
   172     BM     2 35328 14
   173     BM     2 42342 31 32 33 42 58 62
   174     BM     2 54119 01
   175     BM     2 54200 01
   176     BM     2 54201 01
   177     BM     2 94005 33 58 71 72
   178     BM     1 11514 33 42 58 61 72 81
   179     BM     1 34016 33 58 82
   180     BM     1 34025 33 57 59 61
   181     BM     1 34026 33 42 59 61 82
   182     BM     1 34028 33 41 59 61 82
   183     BM     1 34029 41 57 58 61
   184     BM     1 34030 41 57 58 61
   185     BM     1 34035 52 54 56 58
   186     BM     1 34055 53 58
   187     BM     1 34056 42 58 61 82
   188     BM     1 34062 53 72
   189     BM     1 34063 53 82
   190     BM     1 34418 33 82
   191     BM     1 35350 33 82
   192     BM     1 35351 01 02 33
   193     BM     1 44042 32 33
   194     BM     1 54652 04
   195     BM     C 11515 33 59 61 72 82
   196     BM     C 16060 33 41 42 59
   197     BM     C 34022 33 41 57 59 61
   198     BM     C 34023 33 41 42 59 61
   199     BM     C 34057 53
   200     BM     C 34065 53
   201     BM     C 34074 42 58 61
   202     BM     C 34075 58
   203     BM     C 34078 42 55 58 61 91
   204     BM     C 34079 42 58
   205     BM     C 34081 53 58
   206     BM     C 54658 11
```

```
              PAGE NUMBER :    4

              SOURCE CARD LISTING FOR
              OFFLINE READ  ETAM TDSCR

207     BM    CS11516 33 42 43 58 61 82
208     BM    CS11517 53 58 61
209     BM    CS34076 33 41 42 59 61 72 82
210     BM    CS34077 33 41 58 61 72
211     BM    CS34080 33 41 42 59 61 72
212     BM    CS35476 31 43
213     BM    CS35478 11 13
214     BM    CS35479 11
215     BM    CS35480 32 33 34 82
216     BM    CS44375 32 33 34
217     BM    CM34082 33 41 42 57 59 61 72 73
218     BM    CM34083 33 41 42 57 61 72 73
219     BM    CM35483 01 02 14 33 82
220     BM    CM35489 13 82
221     BM    CM35490 13 82
222     BM    CM35492 12 34
223     BM    CM50045 33 34 55 58
224     BM    CM52298 13 14
225     AN      20350 41 42 57 61 73
226     AN      20351 53
227     AN      20352 54
228     AN      20368 54
229     AN      24425 59
230     AN      25352 53
231     AN      25353 53
232     AN      25354 53
233     AN      40490 81
234     AN      42353 52 54
235     AN      46410 41 55
236     AN      48330 53 54
237     AN      62350 53
238     AN      62351 53 54 55
239     AN      62352 53
240     AN      62353 53
241     AN      62354 54
242     AN      62355 54
243     AN      62357 55
244     AN      62358 54
245     AN      62359 52 54
246     AN      62360 54
247     AN      62361 60
248     AN      62362 43 57 61
249     AN      62363 59
250     AN      62364 52 61
251     AN      62365 41 58 62
252     AN      62366 41 57 59
253     AN      62367 53 54
254     AN      62368 53
255     AN      62369 54
256     AN      62370 59 61
257     AN      62371 57 59 61
258     AN      62372 59 61
259     AN      62376 53
260     AN      79350 52 54
261     AN      79356 41 59
262     AN      79357 41 59 72
263     AN      94554 42 57 59 61
264     AN      94555 41 57 61
265     AN      94556 41 59 61
266     AN      94557 54 59
267     AN      99350 41 59 61
268     AN      99351 41 59 61
269     AN      99352 41 59 61
270     AN      99353 53
271     SM    3 34009 41 59
272     SM    3 42020 41 42 62
273     SM    3 42021 41 58
274     SM    3 50303 01
275     SM    3 50928 01
```

```
PAGE NUMBER :    5

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDSCR

276     SM    3 54722 01
277     SM    3 72001 41 52 59 61
278     SM    3 77264 02
279     SM    3 84225 53 55 60 61
280     SM    3 86015 53 55 60 61
281     SM    3 86016 55 60 61
282     SM    3 86017 55 60 61
283     SM    3 86018 55 60 61
284     SM    3 86021 55 61
285     SM    3 86022 41 60
286     SM    3 86023 41 54
287     SM    3 86024 42 60 62 91
288     SM    3 86025 53
289     SM    3 86031 41 60 82
290     SM    3 86035 41 60
291     SM    3 86036 41 60
292     SM    3 86038 43 60
293     SM    3 86365 52 54 61
294     SM    3 86366 41 42 57 61
295     SM    3 94530 41 59 61
296     SM    2 34007 41 42 58 61
297     SM    2 34008 53 55
298     SM    2 38654 01 02
299     SM    2 40469 61 81
300     SM    2 42023 42 58 61
301     SM    2 46036 02 12
302     SM    2 50632 11
303     SM    2 54200 01
304     SM    2 86020 60 81
305     SM    2 86033 53 55 60
306     SM    2 86044 59
307     SM    2 86045 02
308     SM    2 86367 56 60 61
309     SM    2 86368 56 60 61
310     SM    2 86369 56 60 61
311     SM    1 38657 01 11
312     SM    1 38659 31
313     SM    1 44367 34
314     SM    1 44450 32 54
315     SM    1 44451 32 60
316     SM    1 44452 32 41 53
317     SM    1 50986 12
318     SM    1 50987 11
319     SM    1 54057 04
320     SM    1 54725 02
321     SM    1 54726 05
322     SM    1 77266 33 34
323     SM    1 86027 41 53
324     SM    1 86028 33 41 42 60 61
325     SM    1 86029 04
326     SM    1 86034 11 60
327     SM    1 86370 41 55 60
328     SM    1 86371 41 55 60
329     SM    1 86372 41 55 60
330     SM    C 50988 14
331     SM    C 54658 04
332     SM    C 77267 32 33
333     SM    C 86039 33 41 51 52 53 54 58
334     SM    C 86373 33 41 72
335     SM    C 86374 33 41 72
336     SM    CS25371 12
337     SM    CS35476 11 12
338     SM    CS35478 14
339     SM    CS35479 11
340     SM    CS35480 13 14
341     SM    CS35496 35
342     SM    CS35497 34
343     SM    CS35498 14 33 34
344     SM    CS44091 32
```

```
          PAGE NUMBER :    6

          SOURCE CARD LISTING FOR
          OFFLINE READ  ETAM TDSCR

345   SM   CS44375 14 32 34
346   SM   CS50989 33 34
347   SM   CS77268 33 34
348   SM   CS86030 04
349   SM   CM35483 14 33 58
350   SM   CM35489 12 13 14
351   SM   CM35490 12 13 14
352   SM   CM35492 12 13
353   SM   CM52298 14 33
354   EN   3 11088 54 61
355   EN   3 11089 41 61
356   EN   3 11090 41 57 61
357   EN   3 11091 41 52 54 61
358   EN   3 11092 42 57 61
359   EN   3 15329 42 52 54 61 81
360   EN   3 15330 41 57 61
361   EN   3 15331 41 59 61
362   EN   3 18463 41 57 61
363   EN   3 24567 41 57 61
364   EN   3 28085 02 42 51 53 55 58
365   EN   3 30317 41 54 56 57 61
366   EN   3 30313 42 54 56 61 71
367   EN   3 30321 41 57 61
368   EN   3 30323 42 52 54 59 61
369   EN   3 30328 41 55 61
370   EN   3 30336 41 61
371   EN   3 30337 41 61
372   EN   3 30345 41 57 61
373   EN   3 30349 41 57 61
374   EN   3 30353 41 56 61
375   EN   3 30354 41 59 61
376   EN   3 30355 41 57 61
377   EN   3 30357 41 61
378   EN   3 30361 41 54 59 61
379   EN   3 30363 41 54 59 61
380   EN   3 30364 41 54 59 61
381   EN   3 30378 41 57 59 61 81
382   EN   3 30445 41 57 59 61
383   EN   3 31011 41 52 54 56 57 61
384   EN   3 31013 41 57 61
385   EN   3 31014 41 57 61
386   EN   3 31015 41 57 61
387   EN   3 31016 41 57 61
388   EN   3 31017 41 57 61
389   EN   3 31018 41 57 59 61
390   EN   3 31019 01 41 53
391   EN   3 31021 01
392   EN   3 31024 41  52 54 59 61
393   EN   3 31026 41 57 61
394   EN   3 31027 41 57 61
395   EN   3 31028 41 57 61
396   EN   3 31029 41 52 54 61
397   EN   3 31030 41 52 54 61
398   EN   3 31033 41 54 61
399   EN   3 31034 41 57 61
400   EN   3 31035 41 59 61
401   EN   3 31048 41 57 59 61
402   EN   3 31057 41 54 56 81
403   EN   3 31059 54 59 61
404   EN   3 31072 55 61
405   EN   3 31079 42 57 59 61 81
406   EN   3 31091 52 54 61
407   EN   3 31447 54 59 61
408   EN   3 40529 81
409   EN   3 46113 02 55
410   EN   3 51011 01
411   EN   3 51026 01
412   EN   3 54800 53 54
413   EN   3 94155 41 57 61
```

```
PAGE NUMBER :    7

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDSCR

414   EN   3 98024 42 58 61
415   EN   3 98025 71
416   EN   3 98026 42 82
417   EN   3 98335 42 82
418   EN   2 11093 52 54 61 81
419   EN   2 11447 54 72
420   EN   2 30319 41 57 61
421   EN   2 30324 42 57 61
422   EN   2 30325 52 54
423   EN   2 30326 41 52 54 61
424   EN   2 30330 42 52 54 57 61
425   EN   2 30331 41 59 61
426   EN   2 30332 41 54 57 61
427   EN   2 30333 41 54 57 61
428   EN   2 30334 41 54 57 61
429   EN   2 30338 41 54 57 61
430   EN   2 30339 54 61
431   EN   2 30342 41 54 59 61
432   EN   2 30343 41 52 54 61
433   EN   2 30344 41 57 59 61
434   EN   2 30347 42 52 54 61 81
435   EN   2 30358 41 59 61
436   EN   2 30372 41 57 59 61 81
437   EN   2 30374 61 71
438   EN   2 30911 41 52 54 57 61
439   EN   2 30912 41 52 54 61
440   EN   2 30913 41 57 61
441   EN   2 31038 41 52 54 57 59 61 81
442   EN   2 31039 42 57 59 61
443   EN   2 31043 41 52 54 61
444   EN   2 31044 41 57 59 61 81
445   EN   2 31049 42 61 71 81
446   EN   2 31051 61 71
447   EN   2 31061 54 56 58 61
448   EN   2 31092 41 52 54 57 61
449   EN   2 31093 54 61
450   EN   2 31448 41 57 61 71
451   EN   2 31449 41 57 61 71
452   EN   2 31450 41 57 59 61 81
453   EN   2 31451 41 52 54 61 81
454   EN   2 31452 41 57 61
455   EN   2 31453 01 11 12
456   EN   2 35112 01
457   EN   2 35342 01
458   EN   2 48245 61 71 72
459   EN   2 48256 41 52 54 61
460   EN   2 51114 01 11 12
461   EN   2 54801 01 53 54
462   EN   2 54802 01
463   EN   2 94156 41 52 54 57 61
464   EN   2 94157 42 57 81
465   EN   1 11454 33 42 57 58 61 72
466   EN   1 30320 41 52 54 56 58 61 81
467   EN   1 30327 41 59 61
468   EN   1 30339 41 52 54 56 57 61 81
469   EN   1 30340 41 52 54 59 61
470   EN   1 30348 41 52 54 59 61 81
471   EN   1 30350 41 52 54 57 61
472   EN   1 30914 41 52 54 57 59 61 81
473   EN   1 30915 41 52 54 57 59 61 81
474   EN   1 31012 41 42 58 61 71
475   EN   1 31025 41 57 61
476   EN   1 3103[illegible] 12 41 58 61
477   EN   1 31037 61 81
478   EN   1 31040 41 52 54 61
479   EN   1 31041 41 52 54 61
480   EN   1 31042 41 52 54 57 61
481   EN   1 31094 53 55 61
482   EN   1 31097 55 56 61 71
```

```
          PAGE NUMBER :    8

          SOURCE CARD LISTING FOR
          OFFLINE READ  ETAM TDSCR

483     EN    1 31253 41 52 54 57 61
484     EN    1 31454 41 52 54 59 61 81
485     EN    1 35323 01 02
486     EN    1 51167 12
487     EN    1 51168 14
488     EN    1 54827 01 04
489     EN    C 30359 41 53 54 61
490     EN    C 30674 32 33 41 57 59 61 81
491     EN    C 31045 41 52 54 57 61
492     EN    C 31046 41 52 54 57 61
493     EN    C 31047 41 54 57 58 61
494     EN    C 31077 11 41 52 54 61
495     EN    C 31095 41 52 54 56 61
496     EN    C 35432 01 33 42 58 61 82
497     EN    C 35433 12
498     EN    C 35885 31 32 33 41 61
499     EN    C 35886 12
500     EN    C 42278 31
501     EN    C 42279 33 41 42 61
502     EN    C 42367 31
503     EN    C 51069 61 82
504     EN    C 51169 02 14
505     EN    C 51475 12
506     EN    C 51476 55 58 71
507     EN    C 51477 55 71
508     EN    C 51478 55 71
509     EN    C 54804 11 13
510     EN    C 98021 33 42 58 61 71 72
511     EN    CS31410 42 58 60 82
512     EN    CS31455 42 55 61 71
513     EN    CS31456 58 61 71 82
514     EN    CS35072 01 02 33
515     EN    CS35362 01 02 04 31 32
516     EN    CS35476 11 13
517     EN    CS35478 11 13 14
518     EN    CS35479 11
519     EN    CS35480 14 33 34
520     EN    CS35884 12 13 14
521     EN    CS35887 12
522     EN    CS44375 13 14 32 34
523     EN    CS51073 33
524     EN    CS51206 33 82
525     EN    CS98280 12
526     EN    CS98281 12
527     EN    CM20060 13 14 33
528     EN    CM31457 12
529     EN    CM35365 11
530     EN    CM35366 04 11 12 13
531     EN    CM35483 12 14 33
532     EN    CM35489 12 14
533     EN    CM35490 12 13 14
534     EN    CM35492 12 14
535     EN    CM46616 12
536     EN    CM51072 12
537     EN    CM51075 12
538     EN    CM51179 33
539     EN    CM52298 12 13 14
540     EN    CM54806 42 61 71 82
541     GMG   3 14086 41 53 61
542     GMG   3 14564 53 61
543     GMG   3 14565 41 53 61
544     GMG   3 14570 53 54 61
545     GMG   3 15305 41 52 53 54 59 61
546     GMG   3 15307 41 53 54 55 56 60 61
547     GMG   3 15314 54 55 58 61
548     GMG   3 15315 41 52 54 59 61
549     GMG   3 16027 53 54 55 56 61
550     GMG   3 16033 42 57 59 61
551     GMG   3 16034 01 41 59 61
```

```
PAGE NUMBER :    9

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDSCR

552     GMG   3 16304 42 57 61 91
553     GMG   3 16334 41 52 54 59 61
554     GMG   3 16398 41 57 59 61
555     GMG   3 16399 41 57 59 61
556     GMG   3 16400 41 42 57 61 91
557     GMG   3 16402 41 42 59 61
558     GMG   3 16403 41 53 57 59 61
559     GMG   3 16404 41 53 60 61
560     GMG   3 16405 41 42 57 61
561     GMG   3 16407 41 52 54 61
562     GMG   3 16408 42 57 59 61
563     GMG   3 17079 41 52 54 59 61
564     GMG   3 17081 41 52 54 59 61
565     GMG   3 17365 41 52 54 61
566     GMG   3 18017 41 57 61
567     GMG   3 18022 41 53 55 57 61
568     GMG   3 24023 41 42 53 55 57 61
569     GMG   3 24025 41 59 61
570     GMG   3 24390 53 61
571     GMG   3 24391 53 54 61
572     GMG   3 24497 53 54 61
573     GMG   3 24500 41 59 61
574     GMG   3 24501 42 59 81
575     GMG   3 24502 52 54 59 61
576     GMG   3 24503 41 53 61
577     GMG   3 28016 53 61
578     GMG   3 28019 53 61
579     GMG   3 28020 55
580     GMG   3 29236 41 57 59 61
581     GMG   3 29238 41 42 59 61 81
582     GMG   3 29239 41 59 61
583     GMG   3 29549 41 57 59 61
584     GMG   3 40525 57 59 61 81
585     GMG   3 46030 02 55 61
586     GMG   3 48241 01 41 59 61
587     GMG   3 51011 01
588     GMG   3 51026 01
589     GMG   3 54192 01 41 59
590     GMG   3 54759 01 42 59 60
591     GMG   3 54760 53 54 61
592     GMG   3 54761 01 41 53 54 61
593     GMG   3 79272 54 59 61
594     GMG   3 92336 42 57 61 81
595     GMG   3 94424 42 57 59 61 81
596     GMG   3 94508 57 61
597     GMG   2 14043 41 57 61
598     GMG   2 14566 61 71
599     GMG   2 14567 41 53 55 61
600     GMG   2 14798 41 53 61
601     GMG   2 15023 41 53 57 61
602     GMG   2 15309 41 59 61
603     GMG   2 15311 41 42 59 61 71 81
604     GMG   2 15312 41 54 59 61
605     GMG   2 16004 42 56 59 61 72
606     GMG   2 16021 41 52 54 61
607     GMG   2 16022 32
608     GMG   2 16054 42 59 61 72
609     GMG   2 16099 41 51 52 53 54 61
610     GMG   2 17366 53 55 57
611     GMG   2 17367 41 59 61
612     GMG   2 18419 41 57 59 61
613     GMG   2 24026 41 53 61
614     GMG   2 24027 41 57 61
615     GMG   2 24393 61 71
616     GMG   2 28017 42 57 58 61 82
617     GMG   2 28025 53 55
618     GMG   2 28362 26
619     GMG   2 29237 41 53 54 61
620     GMG   2 29243 41 52 54 59 61
```

```
PAGE NUMBER :  10

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM TDSCR

621   GMG  2 29245 41 57 59 61
622   GMG  2 29246 41 52 54 56 61
623   GMG  2 29248 41 54 56 61
624   GMG  2 29249 41 54 56 61
625   GMG  2 29250 41 52 54 61
626   GMG  2 29251 41 54 56 61 81
627   GMG  2 29252 01 02 12
628   GMG  2 36450 01 02
629   GMG  2 36451 01 02
630   GMG  2 51114 01 11
631   GMG  2 54708 01
632   GMG  2 54813 01
633   GMG  2 54814 01 53 54
634   GMG  2 54815 01
635   GMG  1 14062 61 71
636   GMG  1 15313 41 51 52 53 54 57 61 81
637   GMG  1 16023 33 42 61
638   GMG  1 18018 61 71
639   GMG  1 18021 41 53 55 61
640   GMG  1 28018 42 53 55 57 58 82
641   GMG  1 29253 41 52 54 57 61
642   GMG  1 29254 41 54 61 81
643   GMG  1 29255 41 54 57 61 81
644   GMG  1 29256 41 54 57 61 81
645   GMG  1 29551 41 54 57 61 81
646   GMG  1 29552 41 54 57 59 61 81
647   GMG  1 38344 01 12
648   GMG  1 38345 01 11 12
649   GMG  1 38346 01 02
650   GMG  1 38887 01 02
651   GMG  1 38888 14 33
652   GMG  1 51167 12
653   GMG  1 51168 14
654   GMG  1 54816 01 12 14
655   GMG  1 54813 11 12
656   GMG  1 54844 02 33
657   GMG  1 98255 11
658   GMG  C 14042 42 51 52 53 54 58 61 71
659   GMG  C 14044 33 34 55 58 61 72
660   GMG  C 14045 51 52 53 54 55 56 61 72
661   GMG  C 14064 12 14
662   GMG  C 16032 33 61
663   GMG  C 16411 33 61
664   GMG  C 17426 71
665   GMG  C 17427 42 51 52 53 54 55 56 58 61
666   GMG  C 20276 33
667   GMG  C 38892 01 11
668   GMG  C 38893 33
669   GMG  C 38897 12
670   GMG  C 51079 14 33
671   GMG  C 51169 14
672   GMG  C 54766 01 53 54 61
673   GMG  C 98336 33
674   GM   CS17417 33 61
675   GM   CS35341 11 12
676   GM   CS35476 12 13 35
677   GM   CS35478 13 14
678   GM   CS35479 11
679   GM   CS35480 14 33 34
680   GM   CS44375 12 14 32
681   GM   CS51153 12 14
682   GM   CS51154 33 34
683   GM   CS51155 12 13 14
684   GM   CS51206 12 14 33
685   GM   CS54196 33
686   GM   CM20291 33
687   GM   CM35483 14 33
688   GM   CM35489 12 14
689   GM   CM35490 12 14
```

```
            PAGE NUMBER :  11

            SOURCE CARD LISTING FOR
            OFFLINE READ  ETAM TDSCR

 690    GM    CM35492 12
 691    GM    CM35551 33 61
 692    GM    CM35552 11 12
 693    GM    CM35553 12 14 33
 694    GM    CM51156 33
 695    GM    CM51157 12
 696    GM    CM52298 14
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM VCDESC

  1   OFFLINE READ  ETAM VCDESC
  2   6920LLC003292 10 12 21 22 35 42 54 65 73 82
  3   6910LLC003318 02 12 21 31 32 33 43 52 61 71 81
  4   6930LLC001008 09 14 21 22 23 24 33 34 45 55 66 71 81
  5   6910LLC001564 02 11 21 33 34 43 53 61 71 81
  6   6930LLC001407 09 11 13 21 23 24 35 45 52 65 71 81
  7   6910LLC002061 03 11 21 32 45 52 61 71 81
  8   6910LLC002062 03 11 21 32 45 52 61 71 81
  9   6910005723079 03 11 21 32 45 52 61 71 81
 10   6910LLC002063 03 11 21 32 45 52 61 71 81
 11   6910LLC003395 03 11 21 32 45 52 61 71 81
 12   6910LLC003396 03 11 21 32 45 52 61 71 81
 13   6910LLC003397 03 11 21 32 45 52 61 71 81
 14   6910LLC003398 03 11 21 32 45 52 61 71 81
 15   6910LLC001403 03 11 21 32 45 52 61 71 81
 16   6910LLC003399 03 11 21 32 45 52 61 71 81
 17   6910LLC003400 03 11 21 32 45 52 61 71 81
 18   6910LLC003401 03 11 21 32 45 52 61 71 81
 19   6910LLC003402 03 11 21 32 45 52 61 71 81
 20   6910LLC003403 03 11 21 32 45 52 61 71 81
 21   6910LLC000468 03 11 21 32 45 52 61 71 81
 22   6910LLC000469 03 11 21 32 45 52 61 71 81
 23   6910LLC000470 03 11 21 32 45 52 61 71 81
 24   6910LLC001170 03 11 21 32 45 52 61 71 81
 25   6910LLC000471 03 11 21 32 45 52 61 71 81
 26   6910LLC000472 03 11 21 32 45 52 61 71 81
 27   6910005231416 03 11 21 32 45 52 61 71 81
 28   6910LLC000473 03 11 21 32 45 52 61 71 81
 29   6910LLC000474 03 11 21 32 45 52 61 71 81
 30   6940LLC000371 09 13 14 21 32 45 51 54 66 72 81
 31   6940006641435 09 14 21 33 42 55 66 72 83
 32   6910LLC003636 05 12 21 35 43 51 61 71 81
 33   6910LLC003665 03 13 21 36 42 52 61 71 81
 34   6910LLC001614 05 11 21 32 42 52 61 71 81
 35   6910LLC001615 05 11 21 32 42 52 61 71 81
 36   6910LLC001616 05 11 21 32 42 52 61 71 81
 37   6910LLC001617 05 11 21 32 42 52 61 71 81
 38   6910LLC003437 05 11 21 32 42 52 61 71 81
 39   6940LLC003443 05 11 21 32 42 52 61 71 81
 40   6910LLC003444 05 11 21 32 42 52 61 71 81
 41   6940LLC003445 05 11 21 32 42 52 61 71 81
 42   6940LLC003447 05 11 21 32 42 52 61 71 81
 43   6940LLC003448 05 11 21 32 42 52 61 71 81
 44   6940LLC003449 05 11 21 32 42 52 61 71 81
 45   6940005271901 05 11 21 32 42 52 61 71 81
 46   6940005271904 05 11 21 32 42 52 61 71 81
 47   6910005271905 05 11 21 32 42 52 61 71 81
[illegible]   6910005606623 05 11 21 32 42 52 61 71 81
[illegible]   6940005606627 05 11 21 32 42 52 61 71 81
[illegible]   6910005606628 05 11 21 32 42 52 61 71 81
 51   6940005723015 05 11 21 32 42 52 61 71 81
 52   6910005723016 05 11 21 32 42 52 61 71 81
 53   6910005723017 05 11 21 32 42 52 61 71 81
 54   6940005723018 05 11 21 32 42 52 61 71 81
 55   6910006580654 05 11 21 32 42 52 61 71 81
 56   6940007123047 05 11 21 32 42 52 61 71 81
 57   6940008506112 05 11 21 32 42 52 61 71 81
 58   6910LLC001162 05 11 21 32 42 52 61 71 81
 59   6910LLC002134 05 11 21 32 42 52 61 71 81
 60   6920008289165 08 14 21 35 42 52 66 73 81
 61   6920006499282 05 12 21 22 35 42 54 65 71 81
 62   6910LLC000900 03 11 21 32 42 52 61 71 81
 63   6930LLC000593 09 13 21 32 33 45 55 63 72 81
 64   6910LLC002009 02 12 13 14 21 31 32 33 34 41 44 51 54 62 63 64 72 82 83
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
OFFLINE READ  ETAM VDATA

 1  OFFLINE READ  ETAM VDATA
 2  6920LLC0032923C116-3   BOOBY TRAP SIMULATOR, TELEPHONE                          33100
 3  6910LLC0033183A79-1    TRANSPARENCY, M48 TANK: TRACKS, SUSPENSION               84200
 4  6930LLC0010082C30-1    F-4B AIRCRAFT COCKPIT PROCEDURES TRAINER              18950000
 5  6910LLC0015641H4-1     AEROLOGY DISPLAY SET                                    264000
 6  6930LLC0014079E6-1     EJECTION SEAT, UNIVERSAL                               3791000
 7  6910LLC002061120D1A    TOTAL DIFFERENTIAL (MATH) TRAINING AID                   11000
 8  6940LLC00037114B218-1  ASW SYSTEM SIMULATOR-JEZEBEL                            640000
 9  694000664143515Z1-1    APS-T3, ULTRASONIC TRAINER
10  6910LLC00363629EX18    FUEL PUMP, AUTOMOTIVE, TRAINING AID                      83100
11  6910LLC00366518F228-1  FLASH BLINDNESS INDOCTRINATION TRAINER                 2711000
12  6940005271901 26A11    ELECTRONIC CIRCUIT TRAINER
13  69200082891653C52E-1   TARGET SCORING MECHANISM, SINGLE POP-UP
14  69200064992823C55C-1   SIMULATOR, SMALL ARMS, FLASH-NOISE
15  6910LLC0009003E34-1    106MM RECOILLESS RIFLE, FIRING MECH. TRANSP.            102000
16  6930LLC00059315H12-1   R-4B, AIRCRAFT TACTICS TRAINER                       151531800
17  6910LLC00200917H2-1    SIMULATOR, GENERAL PURPOSE                             4229000
18  6910LLC0020621 2CD-2   GRADIENT OF A FUNCTION DEMONSTRATOR                      32500
19  691000572307912CD-3    TRIPLE VECTOR PRODUCT DEMONSTRATOR
20  6910LLC00206312CD-4    STOKES THEORM DEMONSTRATOR                               27500
21  6910LLC00339512CD-5    QUINCUNX BINOMINAL EXPANSION DEMONSTRATOR                55000
22  6910LLC00339612CD-6    ORTHOGRAPHIC PROJECTION DEMONSTRATOR                     38000
23  6910LLC00339712CD-7    TRIGONOMETRIC FUNCTION TRAINING AID                     200000
24  6910LLC00339812CD-8    COMPONENT INTEGRATOR DEMONSTRATOR T/A                    80000
25  6910LLC00140312CD-10   TRIHEDRON DEMONSTRATOR TRAINING AID                      20000
26  6910LLC00339912CD-11   CURVE DEMONSTRATOR TRAINING AID                           7500
27  6910LLC00340012CD-13   VECTOR DISTRIBUTION LAWS TRAINING AID                    40000
28  6910LLC00340112CD-14   SAMPLING BOX TRAINING AID                                38000
29  6910LLC00340212CD-15   CYLINDRIC COORDINATES DEMONSTRATOR T/A                   21000
30  6910LLC00340312CD-16   SPHERICAL COORDINATES DEMONSTRATOR T/A                   21000
31  6910LLC00046812CD-17   MATH, VECTOR ACCELERATION DEMONSTRATOR T/A                5000
32  6910LLC00046912CD-18   MATH, NOMOGRAPHIC PROJECTION DEMONSTRATOR T/A           200000
33  6910LLC00047012CD-19   MATH, PROJECTION, DEMONSTRATOR TRAINING AID             150000
34  6910LLC00117012CD-20   POLAR PLANIMETER MATH DEMONSTRATOR                      120000
35  6910LLC00047112CD-21   MATH, ORDER OF DIFFERENTIATION, DEM T/A                 100000
36  6910LLC00047212CD-22   MATH, FINITE ROTATION, DEMONSTRATOR T/A                   1000
37  691000523141612CD-23   CORIOLIS FORCE DEMONSTRATOR TRAINING AID
38  6910LLC00047312CD-24   MATH, DIRECTION COSINE, DEMONSTRATOR T/A                  5000
39  6910LLC00047412CD-25   MATH, SKEW LINE, DEMONSTRATOR TRAINING AID              150000
40  6910LLC00161426A-11    AMPLIFIER 3-STAGE HF LAB CKT ANALYSIS CHASSIS             1800
41  6910LLC00161526A-11    ULTRASONIC, SONAR, LAB CAU-13                            61600
42  6910LLC00161626A-11    VTVM/AM WITH AC/RF PROBE 27                              11200
43  6910LLC00161726A-11    LAB CKT ANALYSIS CHAS GAS TUBE REGULATOR                  2600
44  6910LLC00343726A-11    RADAR/MICROWAVE SYSTEM LECTURE DEM UNIT                 128300
45  6940LLC00344326A-11    LAB CKT ANALYSIS TROUBLESHOOTING KIT, CAU-2               3700
46  6910LLC00344426A-11    VOLTAGE REGULATOR LECTURE DEM UNIT LDU-2                  8500
47  6940LLC00344526A-11    FREQUENCY MODULATION LAB CKT ANALYSIS UNIT               14900
48  6940LLC00344726A-11    COMPUTER, DIGITAL, BINARY READOUT UNIT, 22A              16000
49  6940LLC00344826A-11    V REGULATOR LAB CKT ANALYSIS CHASSIS CAU-3A               3900
50  6940LLC00344926A-11    CMPTR FUND LAB CKT ANALYSIS UNIT CAU-10                  29000
51  6940005271901 26A-11   ELECTRONICS FUNDAMENTALS UNITS EFU-1
52  694000527190426A-11    LABORATORY CIRCUIT ANALYSIS UNIT CAU-1
53  691000527190526A-11    ELEK CKT LECTURE DEMONSTRATION UNIT LDU-1
54  691000560662326A-11    RADAR MICROWAVE SYS/TEST EQPT TETI TRAINER
55  694000560662726A-11    VOLTAGE REGULATOR LAB CKT ANAL UNIT CAU-3
56  691000560662826A-11    FREQUENCY MODULATION LECTURE DEM UNIT LDU-3
57  694000572301526A-11    RADAR/MICROWAVE SYS CKT ANALYSIS UNIT CAU-5
58  691000572301626A-11    SONAR LECTURE DEMONSTRATION UNIT LDU-5
59  691000572301726A-11    SYNCHRO LECTURE DEM UNIT WITH MOTORS LDU-6
60  694000572301826A-11    SONAR LAB CIRCUIT ANALYSIS UNIT CAU-6
61  691000658065426A-11    TRANSISTOR LECTURE DEMONSTRATION UNIT LDU-7
62  694000712304726A-11    TRANSISTOR LAB CIRCUIT ANALYSIS UNIT CAU-7
63  694000850611226A-11    SYNCHRO LAB CKT ANALYSIS UNIT CAU-9
64  6910LLC00116226A-11/8  COMPUTER SYSTEM LECTURE DEM UNIT LDU-8                  531500
65  6910LLC00213426A-11A   RADAR CIRCUIT CHASSIS TRANSISTORIZED                     30000
```

SECTION D.6

SOURCE LISTINGS OF PL/1 PROGRAMS

The following represents the source program listings of the eighteen PL/1 program modules involved in the ETAM project up through the Range-of-Effect search and search results print.

In the pages following, the programs are source-listed in alphabetic order, P1, P2, P3, ..., P20. The order presented immediately below is the logical order of processing.

Program Name	Program Description
P18	Initialization of the Master Reference data set, ETAM/MREF
P16	Initialization of the Descriptor Master Index, ETAM/DIND
P14	Initialization of the Course data file, ETAM/COURSE
P7	Setup of the Course CIN and CDP directories (ETAM/DCIN and ETAM/DCDP, respectively)
P7A	Course Descriptor file load (ETAM/DESC)
P7B	Printer dump of the contents of the Abbreviated Course data base by option: A = All Course data, sorted by CIN B = All Course data, sorted by CDP C = Course descriptor data
P8	Load of the Abbreviated Vehicle data file (ETAM/VEHS), Vehicle Directory (ETAM/DVEH) and Vehicle Descriptor File (ETAM/VDESC).
P8A	Print of Abbreviated Vehicle data base contents, via Option as follows: A = Print all Vehicle records, ordered by Stock Number B = Print Descriptor data assigned to Vehicles
P9	Load the Abbreviated Tasks data file (ETAM/TASKS), Tasks Directory (ETAM/DTASK), and Tasks Descriptor File (ETAM/TDESC).
P9A	Print, under option control, of the contents of the Abbreviated Tasks data base. Options are as follows: A = Print all Task data B = Print Descriptor data assigned to Tasks

Program Name	Program Description
P20	Print, via selection, the exact Descriptor assignments to each Course, Vehicle, or Task of the system. Options are as follows: C - Print Courses V - Print Vehicles T - Print Tasks
P17	Initialize the files associated with a new ETAM project. This includes files named "projectname"/ ID REC REV REJ
P2	Interactive edit of the Range-of-Effects (ROE) search arguments in files projectname/ REC, REV, and REJ
P3	Range-of-Effects (ROE) search module
P5A	Print of raw (un-edited) results of an ROE search
P5B	Interactive edit of the ROE search results
P5C	Print of the ROE extract results
P1	Print contents of any of the files associated with a given project.

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
P1:      /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */


P1:        /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */
           PROC OPTIONS (MAIN) ;

    /*     REVISIONS :
           24/MAR/THU/77 - VERYFIRST IMPLEMENTATION.

           TUE/29/MAR/77 - MODS PRIOR TO TRANSMIT TO NCSS.
    */

DECLARE ( DIND, COURSE, VEHS, TASKS )
               FILE RECORD DIRECT KEYED ENV ( REGIONAL(1) ) ;

DECLARE DTYPE (21)       CHAR (4) STATIC INIT (    /* FIRST 10 ARE       */
              'ID  ', 'EXD ', 'BPT ', 'RKP ',      /* STRICTLY CARDS     */
              'RRPJ', 'RRPK', 'VQAL', 'VARF',
              'SCEN', 'ALTP',
              'REC ', 'REV ', 'REJ ', 'RESU', 'REE ', /* FROM HERE ON*/
              'EXC ', 'EXJ ', 'EXV ', 'CMR ',  /* SPECIAL FORMATTING */
              'TRER', 'FINR' ) ;

DECLARE ( FLAGS(20), IFLAG ) BIT (1) STATIC,
        NAME4            CHAR (4) STATIC,
        ( CARD, DLINE )        CHAR (80) STATIC,
        DL4              CHAR (4) DEF DLINE POS (1),
        F4               CHAR (4)  DEF CARD,
        F2               CHAR (2)  DEF CARD,
        C110             CHAR (10) DEF CARD POS (1),
        FN               CHAR (4)  DEF CARD POS (3),
        FNME             CHAR (8) STATIC,
        WAY (11:21)      LABEL ;

DECLARE  TTLS (21)       CHAR (40) STATIC INIT (
              'ID - PROJECT DESCRIPTION',
              'EXD - EXTRACT DEFAULTS',
              'BPT - BENEFIT PATTERN',
              'RKP - RISK PROFILE',
              'RRPJ - RISK REDUCTION PROJECTS',
              'RRPK - RISK REDUCTION PACKAGES',
              'VQAL - VARIABLES QUALITATIVE',
              'VARF - VARIABLE REFERENCES',
              'SCEN - SCENARIOS',
              'ALTP - ALTERNATE PROJECTS',
              'REC - COURSES ROE SEARCH ARGUMENTS',
              'REV - VEHICLES ROE SEARCH ARGUMENTS',
              'REJ - TASKS ROE SEARCH ARGUMENTS',
              'RESU - ROE TOTAL SEARCH RESULTS',
              'REE - ROE SEARCH RESULTS',
              'EXC - EXTRACT DB - COURSES',
              'EXJ - EXTRACT DB - JOBTASKS',
              'EXV - EXTRACT DB - VEHICLES',
              'CMR - COST MODEL RESULTS',
              'TRER - TREE RESULTS',
              'FINR - FINANCIAL RESULTS' ) ;

DECLARE 1 DINDEX         STATIC,           /* OVERALL DESCRIPTOR INDEX */
          2 DSTART       FIXED BIN,        /* CKF.DESC.INDEX */
          2 DEND         FIXED BIN,
          2 NCAT         FIXED BIN,
          2 NDESC        FIXED BIN,
          2 CATEGORIES (15),
            3 PCAT       FIXED BIN,
            3 SCAT       FIXED BIN,
            3 ECAT       FIXED BIN,
          2 PDESC (111) FIXED BIN,

        COVER (4)        CHAR (80) BASED (PCOV) ;

DECLARE T2               CHAR (62) STATIC INIT (
 PAGE NUMBER :    2

 SOURCE CARD LISTING FOR
  P1:      /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */
'         XXXXXXXXXX / XXXX' ),

        T2TTL          CHAR (40) DEF T2 POS (23),
        T2NAME         CHAR (10) DEF T2 POS (10),

        ( ROUTE, REROUTE )  LABEL ;

/***************************************/

DECLARE DLINE8         CHAR (8)  DEF DLINE POS (8),
        DLINECST       CHAR (36) DEF DLINE POS (34),
        DLINE28        CHAR (4)  DEF DLINE POS (28),
        DLINE33        CHAR (36) DEF DLINE POS (33),
        DLINE36        CHAR (36) DEF DLINE POS (36),
        DLINE40        CHAR (40) DEF DLINE POS (40),
         CARD7         CHAR (70) DEF CARD POS (8),

        NODIR          CHAR (35) STATIC INIT
             ( '** DIRECTORY SEARCH WAS NEGATIVE **' ),

        NOAB           CHAR (36) STATIC INIT
             ( '(NO ABBREVIATED FILE DATA AVAILABLE)' ) ;
DECLARE ( DISP, PTR, DPAGE, I, J, K, L, LT, NC,
             KALL, KDEV, KTYPE, N, NER, NFIRST, NLINE,
             NN, NPAGE, NR, NTYPE, NV,
             NDF, NJ ) FIXED BIN STATIC,

        DTTL           CHAR (23) STATIC INIT
             ( '       PAGE NUMBER : XXXX' ),
        DDPGE          PICTURE 'ZZZ9' DEF DTTL POS (20),

        MAXLTB         FIXED BIN STATIC INIT ( 60 ) ;

DECLARE C1180          CHAR (70) DEF CARD POS (11) ;

DECLARE 1 MAST         STATIC,
          2 REFS (40)  FIXED BIN,

        1 RESULT       BASED (PR),
          2 RTYP       PICTURE '9',
          2 RSTAT      PICTURE '9',
          2 RDSTAT     PICTURE '9',
          2 RESID      CHAR (13),

        1 RESC         BASED (PR),
          ( 2 F1, 2 F2, 2 F3 )    CHAR (1),
          2 RESCIN     CHAR (8),
          2 RESCDP     CHAR (4),

        1 REST         BASED (PR),
          ( 2 F1, 2 F2, 2 F3 )    CHAR (1),
          2 RATE       CHAR (5),
          2 RANK       CHAR (2),
          2 JOBTASK    CHAR (6),

        S16            CHAR (16) STATIC,
        V1 (3)         LABEL ;

/***************************************/

DECLARE 1 ACRS         STATIC,          /* ABBR. COURSE FILE */
          2 SET1,                       /* CKF.ACOURSES */
            3 CDP      CHAR (4),
            3 CIN      CHAR (8),
            3 CST      CHAR (16),
            3 NOBC     CHAR (4),
            3 NEC      CHAR (4),
          2 PC         CHAR (2),
          2 SET2,
PAGE NUMBER :   3

SOURCE CARD LISTING FOR
 P1:      /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */
                  3 RMS        CHAR (3),
                  3 TYCRS      CHAR (2),
                  3 SVC        CHAR (1),
                  3 MI         CHAR (1),
                2 STCD         CHAR (1),
                2 STDTE        PICTURE '(5)9',
                2 SET3,
                  3 TRAPS      CHAR (1),
                  3 TPC        CHAR (5),
                2 ATTR         PICTURE '99V9',
                2 STBK         PICTURE '99V9',
                2 CLEN         PICTURE '999',
                2 THRS         PICTURE '999',
                2 LHRS         PICTURE '999',
                2 CAJB         PICTURE '(5)9V99',
                2 CC80         CHAR (1) ;

      DECLARE 1 CINDER (304) STATIC,          /* CRS CIN DIRECTORY */
                2 CHCIN        CHAR (8),      /* CKF.CRS.DIRCIN */
                2 CINPTR       FIXED BIN,

              LOADCIN          CHAR (760)  BASED (PLCIN),
              COVCIN (4)       CHAR (760)  BASED (PTCIN) ;

      DECLARE 1 CDPDIR (1200) STATIC,         /* CRS CDP DIRECTORY */
                2 CHCDP        CHAR (4),      /* CKF.CRS.DIRCDP */
                2 CDPPTR       FIXED BIN,

              LOADCDP          CHAR (1200) BASED (PLCDP),
              COVCDP (6)       CHAR (1200) BASED (PTCDP) ;

      /****************************************/

      DECLARE 1 AVEHICLES      STATIC,        /* ABBR. VEH FILE */
                2 STOCKN       CHAR (13),     /* CKF.AVEHS */
                2 DEVDESIG     CHAR (9),
                2 DEVNAME      CHAR (47),
                2 DEVCOST      PICTURE '(8)9V99',
                2 CC80         CHAR (1),

              VCOST            FIXED DEC (10,2) STATIC,
              VOVER            CHAR (80) BASED (PV80) ;

      DECLARE 1 VEHDIR (200)  STATIC,         /* VEHICLE DIRECTORY */
                2 CHVEH        CHAR (13),     /* CKF.VEH.DIR */
                2 VEHPTR       FIXED BIN,

              COVVEH (4)       CHAR (750) BASED (PTVEH),
              LOADVEH          CHAR (750) BASED (PLVEH) ;

      /****************************************/

      DECLARE 1 ATASKS         STATIC,        /* ABBR. TASK FILE */
                2 RATING       CHAR (7),      /* CKF.ATASKS */
                2 JOBTASK      CHAR (6),
                2 TASKTTL      CHAR (50),
                2 FILL         CHAR (6),
                2 BILCOST      PICTURE '(8)9V99',
                2 CC80         CHAR (1) ;

      DECLARE 1 TASKDIR (1200) STATIC,        /* JOBTASK FILE DIRECTORY */
                2 CHTASK       CHAR (13),     /* CKF.TASKS.DIR */
                2 TASKPTR      FIXED BIN,

              COVTASK (12)     CHAR (1500) BASED (PTTASK),
              LOADTASK         CHAR (1500) BASED (PLTASK) ;

      /****************************************/
```

```
PAGE NUMBER :    4

SOURCE CARD LISTING FOR
 P1:      /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */

START:  NTYPE, NC, NJ, NV, NR, NN, NDF,
              NPAGE, DPAGE, LT, NFIRST,
              J, K = 0 ;
        KTYPE = 1 ;
        MAXLTB = 60 ;          /* MAX NUMBER LINES PER TERMINAL PAGE */
        LT = 100 ;             /* TERMINAL LINE COUNTER */
        DISP = 1 ;             /* START OFF WITH DISPLAY ACTIVE */
        PTR = 2 ;              /* START WITH PRINTER INACTIVE */

        FLCIN, FLCDP, FLVEH, FLTASK = '0'B ;

        PCOV = ADDR (DINDEX) ;
        PR = ADDR (S16) ;
        PTCIN = ADDR (CINDER) ;
        PTCDP = ADDR (CDPDIR) ;
        PV80 = ADDR (AVEHICLES) ;
        PTVEH = ADDR (VEHDIR) ;
        PTTASK = ADDR (TASKDIR) ;

COMMENCE:
        OPEN FILE (MREF) RECORD INPUT,
             FILE (ID) RECORD INPUT,
             FILE (COURSE) INPUT,
             FILE (DCIN) RECORD INPUT,
             FILE (DCDP) RECORD INPUT,
             FILE (VEHS) INPUT,
             FILE (DVEH) RECORD INPUT,
             FILE (TASKS) INPUT,
             FILE (DTASK) RECORD INPUT,
             FILE (REE) RECORD INPUT ;

        ON ENDFILE (REE) GO TO EOJ ;

        READ FILE (MREF) INTO (MAST) ;

        READ FILE (ID) INTO (CARD) ;
        T2NAME = C110 ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( 'GENERAL-PURPOSE PROJECT FILE PRINT' ) ;
        DISPLAY ( 'PROGRAM IS NOW STARTING...' ) ;
        DISPLAY ( '     ' ) ;

/****************************************/

   /* THIS IS THE CENTRAL PROMPTING AND DECODE SECTION */

PROMPTA: CARD = ' ' ;
PROMB:  DLINE = '** ENTER FILETYPE FOR OUTPUT, MENU, OR QUIT **' ;

        DISPLAY ( DLINE) REPLY ( CARD ) ;

        J = INDEX ( CARD, ' ' ) ;
        IF ( J = 0 ) THEN GO TO BUMKEY ;

        NAME4 = SUBSTR ( CARD, 1, J ) ;

             KDEV = 0 ;
             REROUTE = PROMPTA ;
LPF:         DO J = 1 TO 21 ;
             KTYPE = J ;
             IF ( NAME4 = DTYPE(J) ) THEN GO TO FINDF ;
             END LPF ;

   /* DISPLY MENU OUTPUT HERE - TO TERMINAL ONLY */

        IF ( NAME4 ¬= 'MENU' ) THEN GO TO TRYQUIT ;

        DLINE = 'PROJECT FILETYPES FOR OUTPUT ARE : ' ;
 PAGE NUMBER :   5
 SOURCE CARD LISTING FOR
  P1:      /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */
        DISPLAY ( '    ' ) ;  DISPLAY ( DLINE ) ;
        DISPLAY ( '    ' ) ;

MOUT:        DO J = 1 TO 21 ;
             DISPLAY ( (5)' ' || TTLS(J) ) ;
             END MOUT ;
                  DISPLAY ( '    ' ) ;
                  GO TO PROMPTA ;

TRYQUIT: IF ( NAME4 = 'QUIT' ) THEN GO TO FEND ;

         IF ( NAME4 = 'ALL' ) THEN GO TO FINDALL ;

BUMKEY: DLINE = '    ** LAST ENTRY UNRECOGNIZED - TRY AGAIN **' ;
        DISPLAY ( '    ' ) ;  DISPLAY ( DLINE ) ;
             GO TO PROMPTA ;

   /* HERE WHEN FOUND A VALID FILE TYPE; START PROCESSING */

FINDF:  IF ( KTYPE > 15 ) THEN
BUMFILE:     DO ;
             DLINE = '    ** FILETYPE ' || NAME4 || ' IS NOT YET '
                  || 'SERVICABLE, TRY AGAIN **' ;

             IF ( PTR = 1 ) THEN
             PUT LIST ( DLINE ) SKIP (2) ;

             DISPLAY ( '    ' ) ;  DISPLAY ( DLINE ) ;
                  DISPLAY ( '    ' ) ;

                  GO TO REROUTE ;
             END BUMFILE ;

   /* NOW, DETERMINE THE VALID OUTPUT DEVICE(S) */

        IF ( KDEV > 1 ) THEN GO TO PROCESS ;
        KDEV = KDEV + 1 ;

        NPAGE = 0 ;
        DISP, PTR = 1 ;

        DISPLAY ( '    ' ) ;
        DISPLAY  ( 'OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER, '
             || 'BOTH, OR QUIT' ) ;

TP2:    DISPLAY ( 'ENTER TERM, PRTR, BOTH, OR QUIT' ) REPLY ( DLINE);
        DISPLAY ( '    ' ) ;

        IF ( DL4 = 'TERM' ) THEN GO TO STERM ;
        IF ( DL4 = 'PRTR' ) THEN GO TO OPRTR ;
        IF ( DL4 = 'BOTH' ) THEN GO TO SPRTR ;
        IF ( DL4 = 'QUIT' ) THEN GO TO FEND ;

        DISPLAY ( '    ' ) ;
        DISPLAY  ( 'UNABLE TO DETERMINE ENTRY TYPE - TRY AGAIN' ) ;
        DISPLAY ( '    ' ) ;

             GO TO TP2 ;

OPRTR:  DISP = 2 ;

SPRTR:  OPEN FILE (SYSPRINT) LINESIZE (120) PAGESIZE (60) ;

        ON ENDPAGE (SYSPRINT)
             BEGIN ;

             NPAGE = NPAGE + 1 ;
             PUT EDIT ( 'PAGE NUMBER : ', NPAGE )
                  ( PAGE, LINE(2), X(9), A, F(4) ) ;
```

```
PAGE NUMBER :    6

SOURCE CARD LISTING FOR
 P1:      /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */

                  IF ( KTYPE > 15 ) THEN GO TO SKP ;

                      PUT EDIT ( 'LISTING OF PROJECT FILE CONTENTS',
                          T2, '      ' )
                          ( SKIP(2), X(9), A, SKIP(1), A, SKIP(1), A ) ;

                  IF ( KTYPE <= 10 ) THEN GO TO SKP ;
                  IF ( KTYPE > 10 ) & ( KTYPE < 14 ) THEN GO TO SKP ;

                  PUT SKIP (1) ;
                  IF ( NTYPE = 3 ) THEN GO TO TASKTTL ;
                  IF ( NTYPE = 2 ) THEN GO TO VEHTTL ;

CRSTTL:           PUT EDIT (
'*** THE FOLLOWING IS EXTRACT DATA FOR COURSES ***',
                      'RECORD',
                      ' NUM    ORIGIN      CIN      COP    COURSE TITLE',
                      '-----  --------   --------  ----  ----------------',
                      '      ' )
                      ( X(9), A,
                        SKIP(2), X(4), A, 3 ( SKIP(1), A ) ) ;
                            GO TO SKP ;

VEHTTL:           PUT EDIT (
'*** THE FOLLOWING IS EXTRACT DATA FOR VEHICLES ***',
                      'RECORD',
                      ' NUM    ORIGIN    STOCK NUMBER  DESIGNATOR',
                      '-----  --------  -------------  ----------',
                      '      ' )
                      ( X(9), A,
                        SKIP(2), X(4), A, COL(35), A, 3 ( SKIP(1), A ) ) ;
                            GO TO SKP ;

TASKTTL:          PUT EDIT (
'*** THE FOLLOWING IS EXTRACT DATA FOR TASKS ***',
                      'RECORD',
                      ' NUM    ORIGIN   RATING  RANK  JOBTASK',
                      '-----  --------  ------  ----  -------', '      ' )
                      ( X(9), A,
                       SKIP(2), X(4), A, 3 ( SKIP(1), A ) ) ;

SKP:              PUT SKIP (1) ;
                  END ;

                      GO TO STARTP ;

STERM:  PTR = 2 ;

STARTP: IF ( DISP = 2 ) THEN GO TO PROCESS ;

      ON CONDITION (NEWDP)
                  BEGIN ;

                  IF ( LT >= 66 ) THEN GO TO LPB ;

                  IF ( LT < 61 ) THEN LT = 61 ;
LPA:                  DO L = LT TO 66 ;
                      DISPLAY ( '      ' ) ;
                      END LPA ;

LPB:              DISPLAY ( '      ' ) ;

                  DPAGE = DPAGE + 1 ;
                  DDPGE = DPAGE ;
                  DISPLAY ( DTTL ) ;
                  DISPLAY ( '      ' ) ;
                  LT = 3 ;
 PAGE NUMBER :    7

 SOURCE CARD LISTING FOR
  P1:     /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */
           IF ( KTYPE > 15 ) THEN GO TO NEWEND ;

           DISPLAY ( 'LISTING OF PROJECT FILE CONTENTS' ) ;
           DISPLAY ( T2 ) ;  DISPLAY ( '     ' ) ;
           LT = LT + 3 ;

           IF ( KTYPE <= 10 ) THEN GO TO NEWEND ;
           IF ( KTYPE > 10 ) & ( KTYPE < 14 ) THEN GO TO NEWEND ;

       IF ( NTYPE = 3 ) THEN DO ;
       DISPLAY ( 'THE FOLLOWING IS ROE RESULT DATA FOR TASKS' ) ;
       DISPLAY (  ' NUM    ORIGIN   RATING  RANK  JOBTASK' ) ;
       DISPLAY (  '-----  --------  ------  ----  -------' ) ;
               GO TO DSKIP ;
           END ;

       IF ( NTYPE = 2 ) THEN DO ;
       DISPLAY ( 'THE FOLLOWING IS ROE RESULT DATA FOR VEHICLES' ) ;
       DISPLAY ( ' NUM    ORIGIN    STOCK NUMBER  DESIGNATOR' ) ;
       DISPLAY ( '-----  --------   ------------  ----------' ) ;
               GO TO DSKIP ;
           END ;

       DISPLAY ( 'THE FOLLOWING IS ROE RESULT DATA FOR COURSES' ) ;
       DISPLAY ( ' NUM    ORIGIN      CIN     CDP    COURSE TITLE' );
       DISPLAY ( '-----  --------   --------  ---- ------------------');

DSKIP:     DISPLAY ( '     ' ) ;
           LT = LT + 4 ;
NEWEND:    END ;

/***************************************/

   /* NOW, NEW FILE IS SETUP */

PROCESS:
       NFIRST = 0 ;
       NR = 0 ;
       FLAGS (KTYPE) = '1'B ;
       T2TTL = TTLS (KTYPE) ;

       OPEN FILE (PROJ) RECORD INPUT  TITLE ( DTYPE(KTYPE) ) ;
       ON ENDFILE (PROJ) GO TO TALLY ;

       IF ( PTR = 1 ) THEN SIGNAL ENDPAGE (SYSPRINT) ;

       IF ( DISP = 2 ) THEN GO TO DOIT ;
           SIGNAL CONDITION (NEWDP) ;

DOIT:  IF ( KTYPE > 10 )
           THEN GO TO WAY ( KTYPE ) ;

/***************************************/

  /* IF CARD-IMAGE PRINTING ALONE REQUIRED, IT IS DONE HERE */

RRR1:  READ FILE (PROJ) INTO (CARD) ;
       NR = NR + 1 ;

       IF ( PTR = 1 ) THEN PUT LIST (CARD) SKIP (1) ;

       IF ( DISP = 2 ) THEN GO TO RRR1 ;
           LT = LT + 1 ;
           IF ( LT > MAXLTB ) THEN SIGNAL CONDITION (NEWDP) ;
           DISPLAY ( CARD ) ;

               GO TO RRR1 ;

/***************************************/
```

```
 PAGE NUMBER :    8

 SOURCE CARD LISTING FOR
  P1:      /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */


    /* IN THIS SECTION, FILETYPES REC, REV, REJ ARE PROCESSED ;
            FIRST, FILETYPE REC IS SETUP */

WAY(11):       DO J = 1 TO 4 ;
               READ FILE (DIND) INTO (CARD) KEY (J) ;
               COVER (J) = CARD ;
               END ;

                    GO TO RECVJ ;

    /* NOW, SETUP FILE (REV) PROCESSING */

WAY(12):       DO J = 5 TO 8 ;
               READ FILE (DIND) INTO (CARD) KEY (J) ;
               COVER (J-4) = CARD ;
               END ;

                    GO TO RECVJ ;

    /* FINALLY, SETUP OF FILE (REJ) ENSUETH */

WAY(13):       DO J = 9 TO 12 ;
               READ FILE (DIND) INTO (CARD) KEY (J) ;
               COVER ( J - 8 ) = CARD ;
               END ;

    /* HERE, ONE OF THE FILES (REC), (REV), (REJ) IS PRINTED */

RECVJ:   READ FILE (PROJ) INTO (CARD) ;
         NR = NR + 1 ;
         IF ( C110 = ' ' ) THEN GO TO RECVJ ;

         GET STRING (C110) EDIT (N) ( F(10) ) ;
         J = ABS (N) ;

         IF ( PDESC(J) = 0 ) THEN GO TO FIXSR ;

         READ FILE (DIND) INTO (CARD)  KEY ( PDESC(J) ) ;

         PUT STRING (DLINE) EDIT ( N, SUBSTR(CARD, 3, 70) )
             ( X(5), F(4), X(1), A ) ;

         IF ( PTR = 1 ) THEN PUT LIST (DLINE) SKIP (2) ;

         IF ( DISP = 2 ) THEN GO TO RECVJ ;

               IF ( LT > MAXLTB ) THEN SIGNAL CONDITION (NEWDP) ;
               DISPLAY ( '    ' ) ;  DISPLAY ( DLINE ) ;
               LT = LT + 2 ;

                    GO TO RECVJ ;

FIXSR:   DLINE = '** ILLEGAL DESCRIPTOR -' || C110 ||
             '- DETECTED IN ' || NAME4 || ' FILE **' ;

         DISPLAY ( '     ' ) ;
         DISPLAY ( (10)' ' || DLINE ) ;
         DISPLAY ( '     ' ) ;
               LT = LT + 3 ;

         PUT LIST ( DLINE ) SKIP (2) ;

               GO TO RECVJ ;

/****************************************/

    /* HERE, EOF SENSED WHILE PROCESSING INPUT FILE */
```

```
PAGE NUMBER :   9

SOURCE CARD LISTING FOR
 P1:     /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */
TALLY:  CLOSE FILE (PROJ) ;

        PUT STRING (DLINE) EDIT ( 'A TOTAL OF ', NR,
             ' RECORDS ARE PRESENT' ) ( X(10), A, F(4), A ) ;
        CARD =    (10)' ' || 'WITHIN THIS FILE' ;

        IF ( PTR = 1 ) THEN PUT EDIT
             ( DLINE, CARD )  ( SKIP(3), A, SKIP(1), A ) ;

        IF ( DISP = 2 ) THEN
             DO ;
             DISPLAY ( '    ' ) ;
             DISPLAY ( 'NOW LISTING FILETYPE = ' || DTYPE (KTYPE) ) ;
                  GO TO TALLY2 ;
             END ;

        DISPLAY ( '    ' ) ;
TALLY2: DISPLAY ( DLINE ) ;
        DISPLAY ( CARD ) ;  DISPLAY ( '    ' ) ;

             GO TO REROUTE ;

/****************************************/

   /* HERE, WHEN PROCECESSING ALL PROJECT FILES ; THE ORDER OF
      PROCESSING WILL BE THAT OF VECTOR (DTYPE(21))    */

FINDALL: REROUTE = TALLYALL ;
        KALL, KTYPE, KDEV = 1 ;

NEXALL: NAME4 = DTYPE (KTYPE) ;

             GO TO FINDF ;

TALLYALL:  IF ( KTYPE >= 21 ) THEN GO TO PROMPTA ;

        KTYPE = KTYPE + 1 ;
        KALL = KALL + 1 ;

             GO TO NEXALL ;

/****************************************/

   /* HEREIN, IS THE PROCESSING OF FILETYPE = REE / RESU */

WAY(14):   WAY(15):
RD1:    READ FILE (PROJ) INTO (S16) ;
        NR = NR + 1 ;

        IF ( RTYP = NTYPE ) THEN GO TO STEP1 ;

             NTYPE = RTYP ;
             IF ( PTR = 1 ) & ( NFIRST > 0 )
                  THEN SIGNAL ENDPAGE (SYSPRINT) ;
             IF ( DISP = 1 ) & ( NFIRST > 0 )
                  THEN SIGNAL CONDITION (NEWDP) ;
             NFIRST = NFIRST + 1 ;

STEP1:  GO TO V1 (NTYPE) ;

   /* HERE, A COURSE RECORD IS PROCESSED */

V1(1):  NC = NC + 1 ;
        NLINE = 1 ;

        DLINE = ' ' ;
        PUT STRING (DLINE) EDIT ( NR, RESCIN, RESCDP )
             ( P'Z,ZZ9', X(12), A, X(2), A ) ;
 PAGE NUMBER :  10

 SOURCE CARD LISTING FOR
  P1:      /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */

STP1:   IF ( RSTAT = 1 ) THEN DLINE8 = 'ORIGINAL' ;
             ELSE IF ( RSTAT = 2 ) THEN DLINE8 = ' ADDED' ;
                                   ELSE DLINE8 = 'DELETED' ;

        IF ( RESCDP = ' ' ) THEN DLINE28 = 'NONE' ;

        IF ( RDSTAT = 2 ) | ( RESCDP = ' ' ) THEN
NOCDATA:     DO ;
             NN = NN + 1 ;
             DLINECST = NOAB ;
                    GO TO OUTLOOP ;
             END NOCDATA ;

        YECDP = RESCDP ;
        LOOKU = STEP2 ;
             GO TO FINDCDP ;

STEP2:  IF ( CDPREC = 0 ) THEN
LP1:         DO ;
             NDF = NDF + 1 ;
             DLINECST = NODIR ;
                       GO TO OUTLOOP ;
             END LP1 ;

        READ FILE (COURSE) INTO (ACRS) KEY (CDPREC) ;

        DLINECST = CST ;

                  GO TO OUTLOOP ;

   /* HERE, VEHICLE TYPE RECORDS ARE PROCESSED */

V1(2):  NV = NV + 1 ;
        NLINE = 1 ;

        DLINE = ' ' ;
        PUT STRING (DLINE) EDIT ( NR, RESID )
             ( P'Z,ZZ9', X(12), A(13) ) ;

STP2:   IF ( RSTAT = 1 ) THEN DLINE8 = 'ORIGINAL' ;
             ELSE IF ( RSTAT = 2 ) THEN DLINE8 = ' ADDED' ;
                                   ELSE DLINE8 = 'DELETED' ;

        IF ( RDSTAT = 2 ) THEN
NOVDATA:     DO ;
             NN = NN + 1 ;
             DLINE33 = NOAB ;
                       GO TO OUTLOOP ;
             END NOVDATA ;

        YEVEH = RESID ;
        LOOKU = STEP3 ;
             GO TO FINDVEH ;

STEP3:  IF ( VEHREC = 0 ) THEN
LP2:         DO ;
             NDF = NDF + 1 ;
             DLINE36 = NODIR ;
                       GO TO OUTLOOP ;
             END LP2 ;

        READ FILE (VEHS) INTO (AVEHICLES) KEY (VEHREC) ;

        DLINE33 = DEVDESIG ;
        CARD = ' ' ;
        CARD7 = DEVNAME ;
        NLINE = 2 ;

                  GO TO OUTLOOP ;
 PAGE NUMBER :  11

 SOURCE CARD LISTING FOR
  P1:      /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */


     /* NOW, PROCESS A TASK RECORD HERE */

V1(3):  NJ = NJ + 1 ;
        NLINE = 1 ;

        DLINE = ' ' ;
        PUT STRING (DLINE) EDIT ( NR, RATE, RANK, REST.JOBTASK )
              ( P'Z,ZZ9', X(12), A, X(4), A, X(3), A ) ;

        IF ( RSTAT = 1 ) THEN DLINE8 = 'ORIGINAL' ;
              ELSE IF ( RSTAT = 2 ) THEN DLINE8 = ' ADDED' ;
                                     ELSE DLINE8 = 'DELETED' ;
STP5:   IF ( RDSTAT = 2 ) THEN
NOTDATA:      DO ;
              NN = NN + 1 ;
              DLINE40 = NOAB ;
                        GO TO OUTLOOP ;
              END NOTDATA ;

        YETASK = RESID ;
        LOOKU = STEP4 ;
              GO TO FINDTASK ;

STEP4:  IF ( TASKREC = 0 ) THEN
LP3:          DO ;
              NDF = NDF + 1 ;
              DLINE40 = NODIR ;
                        GO TO OUTLOOP ;
              END LP3 ;

        READ FILE (TASKS) INTO (ATASKS) KEY (TASKREC) ;

        CARD = ' ' ;
        CARD7 = TASKTTL ;
        NLINE = 2 ;

              GO TO OUTLOOP ;

    /* HERE IS THE PRINTED LINE OUTPUT SEQUENCE */

OUTLOOP: IF ( PTR = 1 )
              THEN PUT LIST (DLINE) SKIP (2) ;

        IF ( PTR = 1 ) & ( NLINE = 2 )
              THEN PUT LIST ( CARD ) SKIP (1) ;

        IF ( DISP = 2 ) THEN GO TO RD1 ;

        IF ( LT > MAXLTB ) THEN SIGNAL CONDITION (NEWDP) ;

        DISPLAY ( '     ' ) ;  DISPLAY ( DLINE ) ;
        LT = LT + 2 ;

        IF ( NLINE ¬= 2 ) THEN GO TO RD1 ;
        DISPLAY ( CARD ) ;
        LT = LT + 1 ;

              GO TO RD1 ;

/**************************************/

DECLARE ( FLCIN, FLCOP, FLVEH, FLTASK )    BIT (1) STATIC,
        YECIN              CHAR (8) STATIC,
        YECDP              CHAR (4) STATIC,
        ( YETASK, YEVEH )  CHAR (13) STATIC,
        ( CINREC, CDPREC, VEHREC, TASKREC, VEHMAX, TASKMAX,
            CINMAX, CDPMAX )    FIXED BIN STATIC,
PAGE NUMBER :  12

SOURCE CARD LISTING FOR
 P1:      /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */

        LOOKU           LABEL ;

   /* CIN DIRECTORY LOOKUP SUBROUTINE

              ENTRY : DESIRED CIN IN YECIN
              EXIT  : CINREC = 0 (NO FIND) OR REG NUMBER
              RETURN VECTOR IS LOOKU          */

FINDCIN: IF ( FLCIN ) THEN GO TO CIN2 ;

        CINMAX = REFS (2) ;
        FLCIN = '1'B ;

CINLOAD:      DO J = 1 TO 4 ;
              PLCIN = ADDR ( COVCIN (J) ) ;
              READ FILE (DCIN) INTO (LOADCIN) ;
              END CINLOAD ;

CIN2:         DO I = 1 TO CINMAX ;
              IF ( YECIN ¬= CHCIN(I) ) THEN GO TO ECIN2 ;
                   CINREC = CINPTR (I) ;
                        GO TO LOOKU ;
ECIN2:        END CIN2 ;

NFCIN:  CINREC = 0 ;
              GO TO LOOKU ;

   /* CDP DIRECTORY LOOKUP SUBROUTINE
              ENTRY : DESIRED CDP IN YECDP
              EXIT  : CDPREC = 0 (NO FIND), OR REC NUMBER
              RETURN VECTOR IS LOOKU          */

FINDCDP: IF ( FLCDP ) THEN GO TO CDP2 ;

        CDPMAX = REFS (3) ;
        FLCDP = '1'B ;

CDPLOAD:      DO J = 1 TO 4 ;
              PLCDP = ADDR ( COVCDP (J) ) ;
              READ FILE (DCDP) INTO (LOADCDP) ;
              END CDPLOAD ;

CDP2:         DO I = 1 TO CDPMAX ;
              IF ( YECDP ¬= CHCDP(I) ) THEN GO TO ECDP2 ;
                   CDPREC = CDPPTR(I) ;
                        GO TO LOOKU ;
ECDP2:        END CDP2 ;

NFCDP:  CDPREC = 0 ;
              GO TO LOOKU ;

   /* VEHICLE ID LOOKUP SUBROUTINE
              ENTRY : DESIRED VEH ID IN YEVEH
              EXIT  : VEHREC = 0 (NO FIND), OR REC NUMBER
              RETURN VECTOR IS LOOKU          */

FINDVEH: IF ( FLVEH ) THEN GO TO VEH2 ;

        VEHMAX = REFS (8) ;
        FLVEH = '1'B ;

VEHLOAD:      DO J = 1 TO 4 ;
              PLVEH = ADDR ( COVVEH (J) ) ;
              READ FILE (DVEH) INTO (LOADVEH) ;
              END VEHLOAD ;

VEH2:         DO I = 1 TO VEHMAX ;
              IF ( YEVEH ¬= CHVEH(I) ) THEN GO TO EVEH2 ;
                   VEHREC = VEHPTR (I) ;
 PAGE NUMBER :  13

 SOURCE CARD LISTING FOR
•  P1:      /* GENERAL-PURPOSE PROJECT FILE PRINT - THU/24/MAR/77 */
                               GO TO LOOKU ;
 EVEH2:          END VEH2 ;

 NFVEH:  VEHREC = 0 ;
                 GO TO LOOKU ;

    /* TASK ID DIRECTORY LOOKJP SUBROUTINE
                 ENTRY : DESIRED TASK ID (13-CHAR) IN YETASK
                 EXIT  : TASKREC = 0 (NO FIND) OR REC NUMBER
                 RETURN VECTOR IS LOOKU          */

 FINDTASK: IF ( FLTASK ) THEN GO TO TASK2 ;

              TASKMAX = REFS (12) ;
           FLTASK = '1'B ;

 TASKLOAD:       DO J = 1 TO 12 ;
                 PLTASK = ADDR ( COVTASK (J) ) ;
                 READ FILE (DTASK) INTO (LOADTASK) ;
                 END TASKLOAD ;

 TASK2:          DO I = 1 TO TASKMAX ;
                 IF ( YETASK ¬= CHTASK(I) ) THEN GO TO ETASK2 ;
                       TASKREC = TASKPTR (I) ;
                            GO TO LOOKU ;
 ETASK2:         END TASK2 ;

 NFTASK: TASKREC = 0 ;
                 GO TO LOOKU ;

 /*****************************************/

    /* IF GET HERE, PROGRAM IS TERRIBLE BROKE */

 BUSTED: PUT STRING (CARD) EDIT
                 ( '       ** ERROR NUMBER ', NER, ' HAS OCCURED **' )
                 ( A, F(2), A ) ;

         PUT LIST (CARD) SKIP (3) ;
         PUT LIST (DLINE) SKIP (2) ;

         DISPLAY ( '     ' ) ;  DISPLAY ( DLINE ) ;
         DISPLAY ( '     ' ) ;

 /*****************************************/

    /* HERE IS EOF WINDUP PROCESSING */

 EOJ:  EOF:
 FEND:    CLOSE FILE (MREF), FILE (ID), FILE (REE),
                FILE (COURSE), FILE (DCDP),
                FILE (VEHS), FILE (DVEH),
                FILE (TASKS), FILE (DTASK) ;

          IF ( PTR = 1 ) THEN CLOSE FILE (SYSPRINT) ;

          DISPLAY ( '     ' ) ;   DISPLAY ( '     ' ) ;
          DISPLAY ( 'PROJECT-FILE PRINT PROGRAM IS NOW TERMINATING' ) ;
          DISPLAY ( '     ' ) ;

                END P1 ;
```

```
PAGE NUMBER :   1

SOURCE CARD LISTING FOR
 P2:      /* SEARCH DESIGNATORS IA EDITOR - TUE/08/FEB/77 */

P2:     /* SEARCH DESIGNATORS IA EDITOR - TUE/08/FEB/77 */
        PROC (INPARM) OPTIONS (MAIN) ;

   /*   REVISIONS :
        TUE/08/FEB/77 - FIRST DEBUG VERSION AT CKF.

        WED/23/FEB/77 - INTRODUCED REC, REV, REJ PROJECT FILES.

        WED/02/MAR/77 - ENTERED CHECKS TO VALIDATE THE INCOMING
                        DESCRIPTORS FROM PROJECT FILES REC,
                        REV, AND REJ; EDIT PROJECT FILES IF INVALID.

        MON/21/MAR/77 - MODS PRIOR TO TRANSMIT TO NCSS.
   */

DECLARE DIND           FILE RECORD DIRECT
             KEYED ENV ( REGIONAL(1) ) ;

DECLARE ( NC, MEOF, KEOF, NRD, J, K, L, FCON, CFLAG, ND,
             JE, KMAX, NER, KD, NTYPE, NUM, L1, L1A, N,
             JD )      FIXED BIN STATIC,

        FNME           CHAR (3) STATIC,
        ROUTE          LABEL,
        ( RD(3), SRET ) LABEL ;

DECLARE ( CARD, DLINE )     CHAR (80) STATIC,
        C14            CHAR (4) DEF CARD POS (1),
        C16            CHAR (6) DEF CARD POS (1),
        C110           CHAR (10) DEF CARD,
        C580           CHAR (75) DEF CARD POS (5),
        C780           CHAR (74) DEF CARD POS (7),
        C1080          CHAR (70) DEF CARD POS (11),

        INPARM         CHAR (100) VARYING,
        1 POVER        BASED (PP),
          2 PARM       CHAR (1),
          2 PARM2      CHAR (4),

        RNUMS (3,3)    FIXED BIN       /* MASTER INDEX REC NOS */
             STATIC INIT ( 1, 4, 0,    /* COURSES */
                           5, 8, 4,    /* VEHICLES */
                           9, 12, 8 ), /* TASKS */

        TYPE3 (3)      CHAR (8) STATIC INIT
             ( 'COURSES', 'VEHICLES', 'TASKS' ),

        FNAMES (3)     CHAR (8) STATIC INIT
             ( 'REC', 'REV', 'REJ' ),

        A80 (80)       CHAR (1) BASED (PA80),
        ( CARG, VARG, TARG ) CHAR (240) STATIC,
        PARGS (3)      POINTER,
        WFLAGS (3)     FIXED BIN STATIC,
        TTL4           CHAR (10) STATIC ;

/*************************************/

DECLARE 1 DINDEX       STATIC,        /* OVERALL DESCRIPTOR INDEX */
          2 DSTART     FIXED BIN,     /* CKF.DESC.INDEX */
          2 DEND       FIXED BIN,
          2 NCAT       FIXED BIN,
          2 NDESC      FIXED BIN,
          2 CATEGORIES (15),
            3 PCAT     FIXED BIN,
            3 SCAT     FIXED BIN,
            3 ECAT     FIXED BIN,
          2 PDESC (111) FIXED BIN,
 PAGE NUMBER :   2

 SOURCE CARD LISTING FOR
  P2:     /* SEARCH DESIGNATORS IA EDITOR - TUE/08/FEB/77 */
        COVER (4)      CHAR (80) BASED (PCOV) ;

DECLARE 1 MAST         STATIC,
          2 REFS (40)  FIXED BIN ;

DECLARE 1 TTLCARD      STATIC,         /* PROJECT TITLE DATA BASE */
          2 PNAME      CHAR (8),       /* CKF.TITLE */
          2 PTTL       CHAR (72),

        1 SARGS        BASED (PSARG), /* ROE SEARCH ARGUMENTS */
          2 SID        CHAR (8),       /* CKF.SEARCH.ARGS */
          2 SSTAT      FIXED BIN,
          2 STAB (115) FIXED BIN ;

/**************************************/

START:  NC, MEOF, KEOF, NRD, NTYPE,
             J, K = 0 ;
        WFLAGS = 0 ;

        SRET = STEP1 ;      /* SET FOR INITIAL SELECT LOOP */

        PARGS(1) = ADDR (CARG) ;
        PARGS(2) = ADDR (VARG) ;
        PARGS(3) = ADDR (TARG) ;
        PP = ADDR (INPARM) ;
        PCOV = ADDR (DINDEX) ;
        PA80 = ADDR (CARD) ;

COMMENCE: OPEN FILE (DIND) INPUT,
             FILE (MREF) RECORD INPUT,
             FILE (ID) RECORD INPUT,
             FILE (SARG) RECORD OUTPUT ;

        READ FILE (MREF) INTO (MAST) ;
        KMAX = REFS (4) ;        /* MAX SS USED IN MASTER INDEX */

FINDT:  READ FILE (ID) INTO (CARD) ;
        TTL4 = C110 ;

        DISPLAY ( '    ' ) ;
        DISPLAY  ( (10)' ' || 'SEARCH DESCRIPTOR EDITOR RUN'
             || ' FOR PROJECT : ' || TTL4 ) ;

        DISPLAY ( C1080 ) ;
        DISPLAY ( '    ' ) ;

/**************************************/

   /* PROJECT FILES REC, REV, REJ TRANSFERRED TO THE SEARCH
      ARGUMENT FILE IN THIS SECTION */

        OPEN FILE (REC) RECORD UPDATE,
             FILE (REV) RECORD UPDATE,
             FILE (REJ) RECORD UPDATE ;

        ON ENDFILE (REC) GO TO L2 ;
        ON ENDFILE (REV) GO TO L3 ;
        ON ENDFILE (REJ) GO TO L4 ;

        ON CONVERSION GO TO FIXSR ;

   /* HERE, COURSES PROCESSED FROM FILE = REC */

        PSARG = ADDR (CARG) ;
        SID = 'COURSES' ;
        SSTAT = 0 ;  STAB = 0 ;
        ROUTE = RW1 ;
        FNME = 'REC' ;
```

```
PAGE NUMBER :    3

SOURCE CARD LISTING FOR
 P2:      /* SEARCH DESIGNATORS IA EDITOR - TUE/08/FEB/77 */

LOOP1:          DO J = 1 TO 4 ;
                READ FILE (DIND) INTO (CARD) KEY ( J ) ;
                COVER (J) = CARD ;
                END LOOP1 ;

R1:         READ FILE (REC) INTO (CARD) ;
            IF ( C110 = ' ' ) THEN GO TO R1 ;

            GET STRING (C110) EDIT (N) ( F(10) ) ;
            J = ABS (N) ;
            IF ( PDESC(J) = 0 ) THEN GO TO FIXSR ;
                SSTAT = SSTAT + 1 ;
                STAB ( ABS(N) ) = N ;
                    GO TO R1 ;

RW1:        REWRITE FILE (REC) FROM (CARD) ;
                GO TO R1 ;

  /* HERE, VEHICLES PROCESSED FROM REV FILE */

L2:         PSARG = ADDR (VARG) ;
            SID = 'VEHICLES' ;
            SSTAT = 0 ;  STAB = 0 ;
            ROUTE = RW2 ;
            FNME = 'REV' ;

LOOP2:          DO J = 5 TO 8 ;
                READ FILE (DIND) INTO (CARD) KEY ( J ) ;
                COVER ( J - 4 ) = CARD ;
                END LOOP2 ;

R2:         READ FILE (REV) INTO (CARD) ;
            IF ( C110 = ' ' ) THEN GO TO R2 ;

            GET STRING (C110) EDIT (N) ( F(10) ) ;
            J = ABS (N) ;
            IF ( PDESC(J) = 0 ) THEN GO TO FIXSR ;
                SSTAT = SSTAT + 1 ;
                STAB ( ABS(N) ) = N ;
                    GO TO R2 ;

RW2:        REWRITE FILE (REV) FROM (CARD) ;
                GO TO R2 ;

  /* FINALLY, TASKS ARE PROCESSED FROM FILE = REJ */

L3:         PSARG = ADDR (TARG) ;
            SID = 'TASKS' ;
            SSTAT = 0 ;  STAB = 0 ;
            ROUTE = RW3 ;
            FNME = 'REJ' ;

LOOP3:          DO J = 9 TO 12 ;
                READ FILE (DIND) INTO (CARD) KEY ( J ) ;
                COVER ( J - 8 ) = CARD ;
                END LOOP3 ;

R3:         READ FILE (REJ) INTO (CARD) ;
            IF ( C110 = ' ' ) THEN GO TO R3 ;

            GET STRING (C110) EDIT (N) ( F(10) ) ;
            J = ABS (N) ;
            IF ( PDESC(J) = 0 ) THEN GO TO FIXSR ;
                SSTAT = SSTAT + 1 ;
                STAB ( ABS(N) ) = N ;
                    GO TO R3 ;

RW3:        REWRITE FILE (REJ) FROM (CARD) ;
```

```
PAGE NUMBER :   4

SOURCE CARD LISTING FOR
 P2:      /* SEARCH DESIGNATORS IA EDITOR - TUE/08/FEB/77 */

                GO TO R3 ;

   /* WITHIN THIS SUBROUTINE, AN INVALID DESIGNATOR (CONVERSION
      OR RANGE) IS ELIMINATED FROM REC, REJ, REV */

FIXSR:  DLINE = '** ILLEGAL DESCRIPTOR -' || C110 ||
              '- DETECTED IN ' || FNME || ' FILE **' ;
        DISPLAY ( '     ' ) ;   DISPLAY ( DLINE ) ;

        DLINE = '   DESCRIPTOR WILL BE IGNORED AND REMOVED FROM THE '
              || FNME || ' FILE' ;
        DISPLAY ( DLINE ) ;
        DISPLAY ( '     ' ) ;

        CARD = ' ' ;
              GO TO ROUTE ;

   /* HERE, FINAL SETUP, PRIOR TO ENTERING MAINLINE */

L4:     CLOSE FILE (REC), FILE (REV), FILE (REJ) ;

        ON CONVERSION
              BEGIN ;

        DISPLAY ( '     ' ) ;
              DISPLAY (
'     ** KEYBOARD ENTRY -' || SUBSTR ( CARD, K, 3 ) ||
'- CONTAINED ILLEGAL NUMERIC CHARACTER' ) ;

                    GO TO PROMPTA ;
              END ;

/**************************************/

   /* HERE, INITIAL LOOP, AWAITING TYPE DESIGNATOR SELECT */

STEP1:  DISPLAY ( '     ' ) ;
        DISPLAY  ( '** MAKE INITIAL ENTRY SELECT' ) REPLY (CARD) ;

        NRD = 1 ;
        IF ( C16 ¬= 'SELECT' ) THEN GO TO STEP1 ;

LP1:          DO J = 1 TO 3 ;
              NTYPE = J ;
              IF ( INDEX ( C780, TYPE3(J) ) > 0 ) THEN GO TO STEP2 ;
              END LP1 ;

        DISPLAY ( '     ' ) ;
        DISPLAY  ( '** SELECTION ERROR - TRY AGAIN' ) ;
              GO TO SRET ;

   /* HERE, A SELECTION HAS BEEN MADE; INDEX IS IN NTYPE */

STEP2:  PSARG = PARGS ( NTYPE ) ;

LP2:          DO J = RNUMS ( NTYPE, 1 ) TO RNUMS ( NTYPE, 2 ) ;
              READ FILE (DIND) INTO (CARD) KEY ( J ) ;
              COVER ( J - RNUMS(NTYPE,3) ) = CARD ;
              END LP2 ;

        DISPLAY ( '     ' ) ;
        DISPLAY  ( '** ' || SID ||
              ' SEARCH DESCRIPTORS SELECTED' ) ;

        IF ( SSTAT = 0 ) THEN
NONEMSG:      DO ;
        DISPLAY ( '     ' ) ;
              DISPLAY  ( '** NO DESCRIPTORS NOW ACTIVE' ) ;
                    GO TO PROMPTA ;
 PAGE NUMBER :    5

 SOURCE CARD LISTING FOR
  P2:     /* SEARCH DESIGNATORS IA EDITOR - TUE/08/FEB/77 */

             END NONEMSG ;

        DISPLAY ( '     ' ) ;
        DISPLAY  ( '** ' || SSTAT ||
             ' DESCRIPTORS NOW SELECTED' ) ;

/**************************************/

   /* CENTRAL ACTION PROMT AND REPLY DECODE FOLLOWS */

PROMPTA: DISPLAY ( '     ' ) ;
         DISPLAY  ( '** MAKE NEXT ACTION REQUEST' ) REPLY ( CARD ) ;

        NRD = 2 ;
        IF ( C14 = 'ADD ' ) THEN GO TO ADD ;
        IF ( C16 = 'PRINT ' ) THEN GO TO PRINT ;
        IF ( C16 = 'DELETE' ) THEN GO TO DELETE ;
        IF ( C16 = 'SELECT' ) THEN GO TO SELECT ;
        IF ( C14 = 'FILE' ) THEN GO TO FILE ;
        IF ( C14 = 'QUIT' ) THEN GO TO QUIT ;

        DISPLAY ( '     ' ) ;
        DISPLAY  ( '** UNIDENTIFIED ACTION REQUEST - TRY AGAIN' ) ;

                  GO TO PROMPTA ;

/**************************************/

   /* HERE IS PROCESSING FOR DELETE OPTION
        ENTRIES ARE - DELETE (BLANK) - MEANS GET RID OF ALL
                                       DESIGNATORS OF SELECTED TYPE.
                    - DELETE N - MEANS DELETE ONLY DESIGNATOR
                                  NUMBERED (N).
   */

DELETE: IF ( C780 = ' ' ) THEN
DELALL:      DO ;                      /* BLANK DELETE SERVICED HERE */
             WFLAGS(NTYPE) = 1 ;
             SSTAT = 0 ;  STAB = 0 ;

        DISPLAY ( '     ' ) ;
             DISPLAY  ( '** ALL ' || SID ||
                  ' SEARCH DESIGNATORS DELETED' ) ;

                       GO TO PROMPTA ;
             END DELALL ;

LP3:         DO J = 8 TO 77 ;     /* LOOK FOR WHERE NUMBER STARTS */
             K = J ;
             IF ( A80(J) ¬= ' ' ) THEN GO TO STEP3 ;
             END LP3 ;

STEP3:  FCON = 1 ;
        GET STRING ( SUBSTR(CARD,K,4) ) EDIT ( NUM ) ( F(4) ) ;

        N = ABS (NUM) ;

        IF ( N > 100 ) | ( N = 0 ) | (PDESC(N) = 0 ) THEN
BUMD:        DO ;
        DISPLAY ( '     ' ) ;
             DISPLAY  ( '**' || NUM ||
                  ' IS NOT A LEGAL DESCRIPTOR FOR ' ||
                  SID || ' TYPE' ) ;
                       GO TO PROMPTA ;
             END BUMD ;

        IF ( STAB(N) ¬= 0 ) THEN GO TO STEP12 ;

        PUT STRING (DLINE) EDIT ( '      ** DESIGNATOR NUMBER ',
PAGE NUMBER :    6

SOURCE CARD LISTING FOR
 P2:      /* SEARCH DESIGNATORS IA EDITOR - TUE/08/FEB/77 */

             N, ' HAS NOT BEEN SELECTED; IT CANNOT BE DELETED' )
             ( A, F(3), A ) ;
        DISPLAY ( DLINE ) ;

             GO TO PROMPTA ;

STEP12: SSTAT = SSTAT - 1 ;
        STAB (N) = 0 ;
        WFLAGS(NTYPE) = 1 ;

        DISPLAY  ( '** ' || SID || ' DESIGNATOR NUMBER' ||
             NUM || ' HAS BEEN DELETED' ) ;

                   GO TO PROMPTA ;

/***************************************/

   /* PRINT OPTION SERVICED IN THIS SECTION */

PRINT:  ND = 0 ;                /* STAB SS */
        CFLAG = 0 ;             /* CATEGORY PRINT FLAG; 0 MEANS DO IT */
        NC = 1 ;                /* CATEGORIES SS */

STEP4:  IF ( ND >= 100 ) THEN GO TO PROMPTA ;

        ND = ND + 1 ;
        IF ( STAB(ND) = 0 ) THEN GO TO STEP4 ;

        JD = STAB (ND) ;    /* FOUND A NON-0 DESIGNATOR */
        KD = PDESC (ND) ;   /* PTR TO DESCRIPTOR CARD */

        IF ( KD <= KMAX ) THEN GO TO STEP5 ;
ER1:         NER = 1 ;
ER1A:        PUT STRING (DLINE) DATA ( ND, JD, KD, KMAX ) ;
                   GO TO BUSTED ;

STEP5:  IF ( ND >= SCAT(NC) ) THEN GO TO STEP7 ;

STEP6:       NC = NC + 1 ;
             CFLAG = 0 ;
             IF ( NC <= 15 ) THEN GO TO STEP5 ;
ER2:               NER = 2 ;
                        GO TO ER1A ;

STEP7:  IF ( ND = SCAT(NC) ) THEN GO TO STEP8 ;
        IF ( ND > ECAT(NC) ) THEN GO TO STEP6 ;
STEP8:  IF ( CFLAG = 1 ) THEN GO TO DETOUT ;

   /* HERE, CATEGORY LINE IS OUTPUT */

CATOUT: READ FILE (DIND) INTO (DLINE) KEY ( PCAT(NC) ) ;
        CFLAG = 1 ;
        DISPLAY ( '     ' ) ;   DISPLAY ( DLINE ) ;

   /* NOW, DETAIL LINE IS OUTPUT */

DETOUT: READ FILE (DIND) INTO (DLINE) KEY ( KD ) ;
        IF ( JD < 0 ) THEN DLINE = '-' || DLINE ;
        DISPLAY ( DLINE ) ;

             GO TO STEP4 ;

/***************************************/

   /* ADD OPTION SETUP HERE */

ADD:         DO J = 5 TO 77 ;
             K = J ;
             IF ( ABO(J) ¬= ' ' ) THEN GO TO STEP9 ;
```

```
                END ADD ;

 STEP9:   FCON = 2 ;
          GET STRING ( SUBSTR(CARD,K,4) ) EDIT (NUM) ( F(4) ) ;

          N = ABS (NUM) ;

          J = PDESC (N) ;

          IF ( N = 0 ) | ( N > 100 ) | ( J = 0 )
                THEN GO TO BUMD ;

          IF ( J <= KMAX ) THEN GO TO STEP10 ;
 ER3:           NER = 3 ;
                JE = SSTAT ;
                PUT STRING (DLINE) DATA ( NUM, J, JE, N ) ;
                     GO TO BUSTED ;

 STEP10: IF ( STAB(N) = 0 ) THEN GO TO STEP13 ;
                STAB (N) = NUM ;
                     GO TO STEP14 ;

 STEP13: STAB (N) = NUM ;
         SSTAT = SSTAT + 1 ;
 STEP14: WFLAGS(NTYPE) = 1 ;

         READ FILE (DIND) INTJ (DLINE) KEY (J) ;

         IF ( NUM < 0 ) THEN DLINE = '-' || DLINE ;

         DISPLAY ( '     ' ) ;
         DISPLAY  ( '** ADDED DESIGNATOR = ' || DLINE ) ;
         DISPLAY  ( '** TOTAL DESIGNATORS SELECTED ARE : '
              || SSTAT ) ;

                    GO TO PROMPTA ;
/****************************************/

   /* SELECT OPTION IMPLEMENTED IN THIS SECTION */

SELECT:       GJ TO LP1 ;

/****************************************/

   /* FILE OPTION IMPLEMENTED HERE */

   /* UPDATED ARRAYS WRITTEN BACK OUT TO REC, REV, REJ PLUS
      SEARCH ARGUMENT FILE */

FILE:         DJ J = 1 TO 3 ;
              PSARG = PARGS (J) ;

              K = J - 1 ;
              WRITE FILE (SARG) FROM (SARGS) ;

              IF ( WFLAGS(J) > 0 ) THEN DO ;
                   DISPLAY ( '     ' ) ;
                   DISPLAY  ( '** MODIFIED ' || SID ||
                        ' DESIGNATORS HAVE BEEN SAVED' ) ;  END ;

              IF ( WFLAGS(J) > 0 ) THEN
LP5:               DO ;

                   OPEN FILE (PROJ) RECORD OUTPUT
                        TITLE ( FNAMES(J) ) ;

                        L1 = SSTAT ;
LP6:                    DO L = 1 TO 100 ;
```

```
                  IF ( STAB(L) = 0 ) THEN GO TO LP6END ;

                  CARD = ' ' ;
                  PUT STRING (CARD) EDIT
                      ( STAB(L) ) ( F(4) ) ;

                  WRITE FILE (PROJ) FROM (CARD) ;

                  SSTAT = SSTAT - 1 ;
LP6END:           END LP6 ;

              L1A = SSTAT ;
              IF ( SSTAT = 0 ) THEN GO TO STEP11 ;
ER4:              NER = 4 ;
                  PUT STRING (DLINE) DATA ( J, L1, L1A ) ;
                      GO TO BUSTED ;

STEP11:       CLOSE FILE (PROJ) ;

              END LP5 ;
          END FILE ;

              GO TO EOJ ;

/***************************************/

  /* THIS IS THE ERROR PROCESSOR */

BUSTED: PUT STRING (DLINE) EDIT
          ( '** ERROR NUMBER ', NER, ' HAS OCCURED **' )
          ( A, F(2), A ) ;
      DISPLAY ( '     ' ) ;  DISPLAY ( DLINE ) ;

          GO TO PROMPTA ;

/***************************************/

  /* THIS IS THE BITTER ENDE */

RD(1):  RD(2):  RD(3):
QUIT:   DISPLAY ( '     ' ) ;
        DISPLAY ( 'SEARCH DESIGNATOR EDITOR IS NOW TERMINATING' ) ;

LP4:          DO J = 1 TO 3 ;
              IF ( WFLAGS(J) = 0 ) THEN GO TO LP4END ;

              DISPLAY ( '     ' ) ;
              DISPLAY (
'** TYPE = ' || TYPE3(J) || ' WAS MODIFIED, BUT NOT SAVED' ) ;

LP4END:       END LP4 ;

EOJ:    CLOSE FILE (DIND), FILE (MREF), FILE (TTL),
              FILE (SARG) ;

        END P2 ;
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
 P3:       /* RANGE-OF-EFFECTS (ROE) SEARCH MODULE - WED/02/MAR/77 */


P3:        /* RANGE-OF-EFFECTS (ROE) SEARCH MODULE - WED/02/MAR/77 */
           PROC OPTIONS (MAIN) ;

    /*     REVISIONS :
           WED/02/MAR/77 - VERYFIRST EDITION.

           SAT/05/MAR/77 - FINAL (HOPEFULLY) DEBUG EFFORTS.

           SUN/20/MAR/77 - MODS PRIOR TO TRANSMIT TO NCSS.
    */

DECLARE ( DIND, COURSE, VEHS,
                TASKS )    FILE RECORD DIRECT KEYED ENV ( REGIONAL (1) ),

           T1              CHAR (38) STATIC INIT (
'RANGE-OF-EFFECTS SEARCH FOR PROJECT : ' ),

           MAXCSS (3)      FIXED BIN STATIC
                INIT ( 2, 8, 12 ),

           MAXRSS (3)      FIXED BIN STATIC
                INIT ( 1, 7, 11 ),

           ( AFIRST, NR, SAVEK, HIT, NOUT, BOT, TOP,
                J, J1, J2, K, K1, K2, L, MAXD, MAXR, N,
                NER, NNR, NTYPE, NNF, MAXD2, KSAVE,
                NC, NF )   FIXED BIN STATIC,

           ( CARD, DLINE )     CHAR (80) STATIC,
           C110            CHAR (10) DEF CARD,
           C1180           CHAR (70) DEF CARD POS (11),
           CRSCIN          CHAR (8) DEF CARD POS (5),
           CRSCDP          CHAR (4) DEF CARD,

           ID8             CHAR (8) STATIC,
           V1 (3)          LABEL ;

DECLARE 1 RESULT           STATIC,           /* ROE SEARCH RESULTS RECORD */
          2 RTYP           PICTURE '9',      /* 1 = CRS, 2 = VEH, 3 = JOB */
          2 RSTAT          PICTURE '9',      /* 1 = ORIGINAL SEARCH,
                                                2 = ADDED, 3 = DELETED    */
          2 RDSTAT         PICTURE '9',      /* 1 = ABBR DATA FOUND,
                                                2 = NO DATA AVAIL           */
          2 RESID          CHAR (13),        /* ENTITY IDENTIFIER */

        1 RESID2           BASED (PRID),     /* OVERLAY FOR COURSE CIN/CDP */
          2 RESCIN         CHAR (8),
          2 RESCDP         CHAR (4),

           DNAMES (3)      CHAR (8) STATIC INIT
                ( 'DESC', 'VDESC', 'TDESC' ),

           CHAR13          CHAR (13) STATIC,
           VJ13            CHAR (13) BASED (P13) ;

DECLARE 1 AND              STATIC,           /* SEARCH ARRAY FOR ANDED-   */
          2 NAND           FIXED BIN,        /* DESCRIPTORS BETWEEN CATS */
          2 TAND (100) FIXED BIN,
          2 CAT (100)  FIXED BIN,

        1 OR (15)          STATIC,           /* SEARCH ARRAY FOR ORED-    */
          2 NOR            FIXED BIN,        /* DESCRIPTORS WITHIN CATS   */
          2 NCAT           FIXED BIN,
          2 TOR (15)       FIXED BIN ;

DECLARE 1 TTLCARD          STATIC,           /* PROJECT TITLE DATA BASE */
          2 PNAME          CHAR (10),        /* CKF.ID */
          2 PTTL           CHAR (70),
 PAGE NUMBER :   2

 SOURCE CARD LISTING FOR
  P3:     /* RANGE-OF-EFFECTS (ROE) SEARCH MODULE - WED/02/MAR/77 */

        1 SARGS        STATIC,        /* ROE SEARCH ARGUMENTS */
          2 SID        CHAR (8),      /* CKF.SEARCH.ARGS */
          2 SSTAT      FIXED BIN,
          2 STAB (115) FIXED BIN,

        RNUMS (3,3)    FIXED BIN       /* MASTER INDEX REC NOS */
             STATIC INIT ( 1, 4, 0,    /* COURSES */
                           5, 8, 4,    /* VEHICLES */
                           9, 12, 8 ); /* TASKS */

/***************************************/

DECLARE 1 CINDESC      STATIC,        /* COURSE DESCRIPTORS */
          2 CINDS      CHAR (8),      /* CKF.CRS.DESC */
          2 FILL       CHAR (5),
          2 DTAB (100) FIXED BIN ;

DECLARE 1 CINDER (304) STATIC,        /* CRS CIN DIRECTORY */
          2 CHCIN      CHAR (8),      /* CKF.CRS.DIRCIN */
          2 CINPTR     FIXED BIN,

        LOADCIN        CHAR (760)  BASED (PLCIN),
        COVCIN (4)     CHAR (760)  BASED (PTCIN) ;

DECLARE 1 DINDEX       STATIC,        /* OVERALL DESCRIPTOR INDEX */
          2 DSTART     FIXED BIN,     /* CKF.DESC.INDEX */
          2 DEND       FIXED BIN,
          2 NCAT       FIXED BIN,
          2 NDESC      FIXED BIN,
          2 CATEGORIES (15),
            3 PCAT     FIXED BIN,
            3 SCAT     FIXED BIN,
            3 ECAT     FIXED BIN,
          2 PDESC (111) FIXED BIN,

        COVER (4)      CHAR (80) BASED (PCOV) ;

DECLARE 1 MAST         STATIC,
          2 REFS (40)  FIXED BIN ;

DECLARE 1 VEHDESC      STATIC,        /* VEH DESCRIPTOR FILE */
          2 VEHDS      CHAR (13),     /* CKF.VEH.DESC */
          2 VTAB (100) FIXED BIN ;

DECLARE 1 VEHDIR (200) STATIC,        /* VEHICLE DIRECTORY */
          2 CHVEH      CHAR (13),     /* CKF.VEH.DIR */
          2 VEHPTR     FIXED BIN,

        COVVEH (4)     CHAR (750) BASED (PTVEH),
        LOADVEH        CHAR (750) BASED (PLVEH) ;
DECLARE 1 TASKDESC     STATIC,        /* TASK DESCRIPTOR FILE */
          2 TASKDS     CHAR (13),     /* CKF.TASK.DESC */
          2 TTAB (100) FIXED BIN ;

DECLARE 1 TASKDIR (1200) STATIC,      /* JOBTASK FILE DIRECTORY */
          2 CHTASK     CHAR (13),     /* CKF.TASKS.DIR */
          2 TASKPTR    FIXED BIN,

        COVTASK (12)   CHAR (1500) BASED (PTTASK),
        LOADTASK       CHAR (1500) BASED (PLTASK) ;

/***************************************/

COMMENCE:
        PRID = ADDR (RESID) ;
        P13 = ADDR (CINDS) ;
        PTCIN = ADDR (CINDER) ;
        PCOV = ADDR (DINDEX) ;
 PAGE NUMBER :    3

 SOURCE CARD LISTING FOR
  P3:      /* RANGE-OF-EFFECTS (ROE) SEARCH MODULE - WED/02/MAR/77 */

        PTVEH = ADDR (VEHDIR) ;
        PTTASK = ADDR (TASKDIR) ;

        OPEN FILE (DIND) INPUT,
             FILE (MREF) RECORD INPUT,
             FILE (ID) RECORD INPUT,

             FILE (SARG) RECORD INPUT,
             FILE (RESU) RECORD OUTPUT,

             FILE (COURSE) INPUT,
             FILE (DCIN) RECORD INPUT,
             FILE (VEHS) INPUT,
             FILE (DVEH) RECORD INPUT,
             FILE (TASKS) INPUT,
             FILE (DTASK) RECORD INPUT ;

   /* FILE DIRECTORIES LOADED IN THIS SECTION */

CINLOAD:     DO J = 1 TO 4 ;
             PLCIN = ADDR ( COVCIN (J) ) ;
             READ FILE (DCIN) INTO (LOADCIN) ;
             END CINLOAD ;

VEHLOAD:     DO J = 1 TO 4 ;
             PLVEH = ADDR ( COVVEH (J) ) ;
             READ FILE (DVEH) INTO (LOADVEH) ;
             END VEHLOAD ;

TASKLOAD:    DO J = 1 TO 12 ;
             PLTASK = ADDR ( COVTASK (J) ) ;
             READ FILE (DTASK) INTO (LOADTASK) ;
             END TASKLOAD ;

   /* NOW, OUTPUT THE PROJECT TITLE */

        READ FILE (ID) INTO (CARD) ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( 'RANGE-OF-EFFECTS (ROE) SEARCH '
             || 'PROGRAM IS STARTING' ) ;

        DLINE = T1 || C110 ;
        DISPLAY ( '     ' ) ;   DISPLAY ( DLINE ) ;

        DLINE = C1180 ;
        DISPLAY ( DLINE ) ;

   /* FINAL SETUP BEFORE PROGRAM MAINLINE */

        NTYPE = 1 ;         /* START WITH COURSES */
        NOUT = 0 ;          /* TOTAL # OUPUT (HIT) RECORDS */
        RSTAT = 1 ;         /* RESULT RECORD = ALL ORIGINAL */

        READ FILE (MREF) INTO (MAST) ;

/***************************************/

   /* THIS IS START OF MAINLINE LOOP */

START:  RTYP = NTYPE ;                /* RESULT RECORD TYPE */
        NC, NF, NNF = 0 ;             /* DESCRIPTOR RECORD COUNTS */

        READ FILE (SARG)  INTO (SARGS) ;

        PUT STRING (DLINE) EDIT ( '** NOW PROCESSING ', SID,
             ' SEARCH - ', SSTAT, ' DESCRIPTORS SELECTED' )
             ( A, A(8), A, F(3), A ) ;
        DISPLAY ( '     ' ) ;   DISPLAY ( DLINE ) ;
 PAGE NUMBER :    4

 SOURCE CARD LISTING FOR
  P3:     /* RANGE-OF-EFFECTS (ROE) SEARCH MODULE - WED/02/MAR/77 */

      IF ( SSTAT <= 0 ) THEN
LP1:      DO ;

          DLINE = '** NO SEARCH DESCRIPTORS SPECIFIED FOR '
              || SID || ' TYPE DATA - SEARCH IS BYPASSED' ;
      DISPLAY ( '     ' ) ;   DISPLAY ( DLINE ) ;

              GO TO TALLY ;
          END LP1 ;

   /* HERE, MASTER DESCRIPTOR INDEX IS LOADED FOR APPROPRIATE NTYPE */

LOADDIND:     DO J = RNUMS(NTYPE,1) TO RNUMS(NTYPE,2) ;
              READ FILE (DIND) INTO (CARD) KEY (J) ;
              COVER ( J - RNUMS(NTYPE,3) ) = CARD ;
              END LOADDIND ;

          NAND = 0 ;
          NOR = 0 ;

          MAXD = REFS ( MAXCSS(NTYPE) ) ;     /* MAX DIRECTORY SS */

          MAXD2 = MAXD / 2 ;  /* HALFWAY POINT FOR TASKS TYPE ONLY */

          MAXR = REFS ( MAXRSS(NTYPE) ) ;     /* TOTAL # DATA RECS */

/***************************************/

   /* NOW, LOAD UP ARRAYS PRIOR TO SEARCH OF DESIGNATOR DATASET.
      FIRST, SETUP THE -OR- ARRAY FOR HANDLING DESIGNATORS
      WITHIN A SINGLE CATEGORY    */

              NOR = 0 ;               /* ALL COUNT FIELDS = 0 */
              L = 0 ;                 /* OR STRUCTURE SS */
LP2:          DO J = 1 TO 15 ;        /* CATEGORY SS */
              AFIRST = 0 ;            /* HAVE NOT STARTED A CAT YET */
              K1 = SCAT (J) ;
              K2 = ECAT (J) ;

LP3:              DO K = K1 TO K2 ;   /* INDIVIDUAL CAT BOUNDS */

                  IF ( STAB(K) = 0 ) THEN GO TO LP3END ;

                  IF ( AFIRST ¬= 0 ) THEN GO TO STEP1 ;
                      AFIRST = 1 ;
                      SAVEK = STAB (K) ;
                      KSAVE = K ;
                          GO TO LP3END ;

STEP1:            IF ( AFIRST < 0 ) THEN GO TO STEP2 ;

                      L = L + 1 ;
                      NOR (L) = 1 ;
                      TOR (L,1) = SAVEK ;
                      STAB (KSAVE) = 0 ;
                      AFIRST = -1 ;
                      OR.NCAT (L) = J ;

STEP2:            NOR(L) = NOR(L) + 1 ;
                  TOR ( L, NOR(L) ) = STAB (K) ;
                  STAB (K) = 0 ;
LP3END:           END LP3 ;

              END LP2 ;

   /* NOW, SETUP THE -AND- STRUCTURE TO ENCOMPASS DESIGNATORS
      THAT ARE TO BE ANDED BETWEEN CATEGORIES */
 PAGE NUMBER :   5

 SOURCE CARD LISTING FOR
  P3:      /* RANGE-OF-EFFECTS (ROE) SEARCH MODULE - WED/02/MAR/77 */

            NAND = 0 ;                    /* AND STRUCTURE COUNTER */
            K = 0 ;                       /* CATEGORY SS, 1 TO 15  */

LP4:        DO J = 1 TO 100 ;             /* STAB SS */

            IF ( STAB(J) = 0 ) THEN GO TO LP4END ;

STEP3:      K = K + 1 ;
            IF ( J >= SCAT(K) ) THEN GO TO STEP4 ;

ER1:            NER = 1 ;
ER1A:           PUT STRING (DLINE) DATA ( NAND, J, K, SCAT(K) ) ;
                    GO TO BUSTED ;

STEP4:      IF ( J > ECAT(K) ) THEN GO TO STEP6 ;

                NAND = NAND + 1 ;
                IF ( NAND <= 100 ) THEN GO TO STEP5 ;

ER2:                NER = 2 ;
                        GO TO ER1A ;

STEP5:          TAND (NAND) = STAB (J) ;
                CAT (NAND) = K ;
                    GO TO LP4END ;

STEP6:      IF ( K < 15 ) THEN GO TO STEP3 ;

ER3:                NER = 3 ;
                        GO TO ER1A ;

LP4END:     END LP4 ;

/*************************************/

   /* HERE WE ARE AT THE SEARCH - FIRST, THE SINGLE DESIGNATORS TO
      BE ANDED TOGETHER BETWEEN CATEGORIES ARE PURSUED */

        OPEN FILE (DESIGS) RECORD INPUT TITLE ( DNAMES(NTYPE) ) ;
        ON ENDFILE (DESIGS) GO TO EOFDESC ;
        NC, NF, NNF = 0 ;

NOGO:
RD1:    READ FILE (DESIGS) INTO (CINDESC) ;
        NC = NC + 1 ;

        RDSTAT = 1 ;             /* ASSUME ABBR. DATA EXISTS */
SEARCH: HIT = 0 ;                /* NOGO, JUST STARTING */

        IF ( NAND = 0 ) THEN GO TO SEARCHOR ;

ANDSEARCH:   DO J = 1 TO NAND ;
             N = ABS ( TAND(J) ) ;
             IF ( TAND(J) < 0 ) THEN GO TO TESTNOT ;

TESTPOS:     IF ( DTAB(N) > 0 ) THEN GO TO GOAND ;
                 GO TO ALTAND ;

TESTNOT:     IF ( DTAB(N) <= 0 ) THEN GO TO GOAND ;
                 GO TO NOGO ;

ALTAND:          K1 = SCAT ( CAT(J) ) ;
                 K2 = ECAT ( CAT(J) ) ;
LP5:             DO K = K1 TO K2 ;
                 IF ( DTAB(K) ¬= 0 ) THEN GO TO NOGO ;
                 END LP5 ;

GOAND:       END ANDSEARCH ;
```

```
PAGE NUMBER :    6

SOURCE CARD LISTING FOR
 P3:     /* RANGE-OF-EFFECTS (ROE) SEARCH MODULE - WED/02/MAR/77 */
SEARCHOR:    HIT = 1 ;       /* HERE, MADE IT ONE WAY OR TOTHER */

   /* NOW, WE TACKLE THE OR SEARCH */

        IF ( L > 0 )          /* L IS COUNT OF OR STRUCTURES */
             THEN GO TO ORSEARCH ;

        IF ( HIT = 0 ) THEN GO TO NOGO ;
                       ELSE GO TO GO ;

ORSEARCH:    DO J = 1 TO L ;      /* OR-STRUCTURE SS */

                  K1 = NOR (J) ;      /* TOT DESCS WITHIN CAT */
LP6:              DO K = 1 TO K1 ;
                  N = TOR (J,K) ;

                  IF ( N > 0 ) THEN GO TO POSTEST ;

NEGTEST:          IF ( DTAB( ABS(N) ) <= 0 ) THEN GO TO ORGO ;
                       GO TO LP6END ;

POSTEST:          IF ( DTAB(N) > 0 ) THEN GO TO ORGO ;

LP6END:           END LP6 ;

             K1 = SCAT ( OR.NCAT(J) ) ;
             K2 = ECAT ( OR.NCAT(J) ) ;

LP7:              DO K = K1 TO K2 ;
                  IF ( DTAB(K) ¬= 0 ) THEN GO TO NOGO ;
                  END LP7 ;

ORGO:        END ORSEARCH ;

   /* IF FALL THRU FROM ABOVE, ALL IS GO */

GO:          GO TO V1 (NTYPE) ;

/**************************************/

   /* HERE, COURSE RESULTS PROCESSED */

V1(1):  J1 = MAXD ;            /* MAX SS USED IN CRS DIRECTORY */

LP8:         DO J = 1 TO J1 ;     /* LOOK UP FOUND CIN */
             K = J ;
             IF ( CINDS = CHCIN(J) )
                  THEN GO TO CRSGO ;
LP8END:      END LP8 ;

CRSNOGO:     /* NO FIND IF FALL OUTTA ABOVE */
        RESID = ' ' ;
        RDSTAT = 2 ;
        RESCIN = CINDS ;

STEP7:  WRITE FILE (RESU) FROM (RESULT) ;
             NOUT = NOUT + 1 ;
             NNF = NNF + 1 ;

                  GO TO RD1 ;

CRSGO:       /* HERE, CRS DIRECTORY SEARCH WAS GO */
        RDSTAT = 1 ;
        K1 = K ;
        K = CINPTR (K) ;

        IF ( K <= MAXR ) THEN GO TO STEP8 ;

ER4:         NER = 4 ;
```

```
 PAGE NUMBER :    7

 SOURCE CARD LISTING FOR
  P3:      /* RANGE-OF-EFFECTS (ROE) SEARCH MODULE - WED/02/MAR/77 */

               PUT STRING (DLINE) DATA ( K, K1, REFS(1), CINDS ) ;
                    GO TO BUSTED ;

STEP8:   READ FILE (COURSE) INTO (CARD) KEY ( K ) ;

         IF ( CINDS ¬= CRSCIN ) THEN GO TO RD1 ;

         RESID = ' ' ;
         RESCIN = CRSCIN ;
         RESCDP = CRSCDP ;

         WRITE FILE (RESU) FROM (RESULT) ;
         NOUT = NOUT + 1 ;
         NF = NF + 1 ;

               K = K + 1 ;
               IF ( K > REFS(1) ) THEN GO TO RD1 ;
                    GO TO STEP8 ;

/**************************************/

   /* HERE, VEHICLE RESULTS ARE OUTPUT */

V1(2):   J1 = MAXD ;            /* MAX SS USED IN VEH DIRECTORY */
         RESID = VJ13 ;

LP9:           DO J = 1 TO J1 ;
               K = J ;
               IF ( VJ13 = CHVEH(J) )
                    THEN GO TO VEHGO ;
LP9END:        END LP9 ;

VEHNOGO:       /* IF FALL OUTTA ABOVE, NO FIND */
         RDSTAT = 2 ;
               GO TO STEP7 ;

VEHGO:         /* HERE, FOUND WANTED VEHICLE IN DIRECTORY */
         RDSTAT = 1 ;

         IF ( VEHPTR(K) <= MAXD ) THEN GO TO STEP9 ;

ER5:           NER = 5 ;
               PUT STRING (DLINE) DATA ( K, VEHPTR(K), REFS(7),
                    RESID ) ;
                    GO TO BUSTED ;

STEP9:   WRITE FILE (RESU) FROM (RESULT) ;
         NOUT = NOUT + 1 ;
         NF = NF + 1 ;

               GO TO RD1 ;

/**************************************/

   /* FINALLY, THE TASK RESULTS ARE PROCESSED */

V1(3):   RESID = VJ13 ;

         IF ( VJ13 >= CHTASK(MAXD2) )
               THEN DO :  BOT = MAXD2 ;  TOP = MAXD ;  END ;
               ELSE DO ;  BOT = 1 ;  TOP = MAXD2 ;  END ;

LP10:          DO J = BOT TO TOP ;
               K = J ;
               IF ( VJ13 = CHTASK(J) )
                    THEN GO TO TASKGO ;
LP10END:       END LP10 ;

TASKNOGO:      /* IF FALL OUTTA ABOVE, NO FIND AT ALL */
```

```
PAGE NUMBER :    8

SOURCE CARD LISTING FOR
 P3:      /* RANGE-OF-EFFECTS (ROE) SEARCH MODULE - WED/02/MAR/77 */
                    GO TO VEHNOGO ;

TASKGO:        /* HERE, TASK DIRECTORY SEARCH WAS GOLDEN */
            RDSTAT = 1 ;

            IF ( TASKPTR(K) <= REFS(11) ) THEN GO TO STEP9 ;

ER6:                NER = 6 ;
                    PUT STRING (DLINE) DATA ( K, TASKPTR(K), REFS(11),
                        RESID ) ;
                        GO TO BUSTED ;

/***************************************/

  /* HERE, WHEN EOF REACHED ON DESCRIPTOR INPUT DATASET */

EOFDESC: CLOSE FILE (DESIGS) ;

            PUT STRING (DLINE) EDIT
                    ( 'TOTAL NUMBER DESCRIPTOR RECORDS EXAMINED : ', NC )
                    ( X(7), A, P'ZZ,ZZ9' ) ;
            DISPLAY ( '     ' ) ;    DISPLAY ( DLINE ) ;

            PUT STRING (DLINE) EDIT
                    ( 'TOTAL RECORDS SELECTED WITH MATCHING DATA : ', NF )
                    ( X(6), A, P'ZZ,ZZ9' ) ;
            DISPLAY ( DLINE ) ;

            PUT STRING (DLINE) EDIT
                    ( 'TOTAL RECORDS SELECTED WITHOUT ABBREVIATED DATA : ',
                      NNF ) ( A, P'ZZ,ZZ9' ) ;
            DISPLAY ( DLINE ) ;

            PUT STRING (DLINE) EDIT
                    ( 'ACCUMULATED SEARCH OUTPUT RECORDS, THUS FAR : ',
                      NOUT ) ( X(4), A, P'ZZ,ZZ9' ) ;
            DISPLAY ( DLINE ) ;

  /* NOW, DETERMINE IF ANOTHER CYCLE IS REQUIRED */

TALLY:  IF ( NTYPE >= 3 ) THEN GO TO EOJ ;
                NTYPE = NTYPE + 1 ;

                    GO TO START ;

/***************************************/

  /* HERE, PROGRAM IS IN STAGE OF TERMINAL ILLNESS */

BUSTED: PUT STRING (CARD) EDIT
                    ( '      ** ERROR NUMBER ', NER, ' HAS OCCURED **' )
                    ( A, F(2), A ) ;
            DISPLAY ( '     ' ) ;
            DISPLAY ( CARD ) ;

            DISPLAY ( '     ' ) ;  DISPLAY ( DLINE ) ;

/***************************************/

  /* THIS IS THE BITTER ENDE */

EOJ:     DLINE = '** DESIGNATOR SEARCH PROGRAM IS TERMINATING' ;
            DISPLAY ( '     ' ) ;  DISPLAY ( DLINE ) ;

            PUT STRING (DLINE) EDIT
                    ( '   TOTAL ACCUMULATED SEARCH OUTPUT RECORDS : ',
                      NOUT ) ( A, P'ZZ,ZZ9' ) ;
            DISPLAY ( '     ' ) ;  DISPLAY ( DLINE ) ;
            DISPLAY ( '    ' ) ;
```

```
PAGE NUMBER :    9

SOURCE CARD LISTING FOR
 P3:       /* RANGE-OF-EFFECTS (ROE) SEARCH MODULE - WED/02/MAR/77 */


          CLOSE FILE (DIND), FILE (COURSE),
                FILE (MREF), FILE (DCIN),
               FILE (SARG), FILE (RESU), FILE (ID) ;

               END P3 ;
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
 P5A:     /* PRINT OF RAW ROE RESULTS FILE - MON/28/FEB/77 */

P5A:     /* PRINT OF RAW ROE RESULTS FILE - MON/28/FEB/77 */
         PROC (INPARM) OPTIONS (MAIN) ;

   /*    REVISIJNS :
         WED/02/MAR/77 - FIRST IMPLEMENTATION.

         SAT/J5/MAR/77 - FINAL (HOPEFULLY) DEBUG EFFORTS.

         THU/24/MAR/77 - MODS PRIOR TO TRANSMIT TO NCSS.
   */

DECLARE 1 TTLCARD        STATIC,          /* PROJECT TITLE DATA BASE */
          2 PNAME        CHAR (10),       /* CKF.ID */
          2 PTTL         CHAR (70),

        INPARM           CHAR (100) VARYING,
        1 POVE3          BASED (PP),
          2 PARM2        CHAR (4),

        ( DISP, PTR, DPAGE, I, J, K, L, LT, NC,
             NN, NPAGE, NR, NTYPE, NV,
             NDF, NJ ) FIXED BIN STATIC,

        DTTL             CHAR (23) STATIC INIT
            ( '      PAGE NUMBER : XXXX' ),
        DDPGE            PICTURE 'ZZZ9' DEF DTTL POS (20),

        MAXLTB           FIXED BIN STATIC INIT ( 60 ) ;

DECLARE VINTAGE (3)      CHAR (8) STATIC INIT
            ( 'COURSES', 'VEHICLES', 'TASKS' ),

        DTTL2            CHAR (39) STATIC INIT
            ( 'RANGE-OF-EFFECTS RESULTS FOR : XXXXXXXX' ),
        D2TY             CHAR (8) DEF DTTL2 POS (32),

        DTTL3            CHAR (39) STATIC INIT ( (39)'-' ) ;

DECLARE ( COURSE, VEHS, TASKS
            ) FILE RECORD DIRECT KEYED ENV ( REGIONAL(1) ) ;

DECLARE ( CARD, DLINE )       CHAR (80) STATIC,
        C110             CHAR (10) DEF CARD,
        C1180            CHAR (70) DEF CARD POS (11) ;

DECLARE 1 MAST           STATIC,
          2 REFS (40)    FIXED BIN,

        1 RESULT         BASED (PR),
          2 RTYP         PICTURE '9',
          2 RSTAT        PICTURE '9',
          2 RDSTAT       PICTURE '9',
          2 RESID        CHAR (13),

        1 RESC           BASED (PR),
          ( 2 F1, 2 F2, 2 F3 )   CHAR (1),
          2 RESCIN       CHAR (8),
          2 RESCDP       CHAR (4),

        1 REST           BASED (PR),
          ( 2 F1, 2 F2, 2 F3 )   CHAR (1),
          2 RATE         CHAR (5),
          2 RANK         CHAR (2),
          2 JOBTASK      CHAR (6),

        S16              CHAR (16) STATIC,
        V1 (3)           LABEL ;
```

```
PAGE NUMBER :    2

SOURCE CARD LISTING FOR
 P5A:     /* PRINT OF RAW ROE RESULTS FILE - MON/28/FEB/77 */

DECLARE T1                CHAR (32) STATIC INIT (
'RECNO    CIN    CDP          COURSE' ),

        T1U               CHAR (37) STATIC INIT (
'------ -------- ---- ----------------' ),

        T2                CHAR (46) STATIC INIT (
'RECNO  STOCK NUMBER  FILE NUM  TRAINING DEVICE' ),

        T2U               CHAR (46) STATIC INIT (
'------ ------------- --------- ---------------' ),

        T3                CHAR (29) STATIC INIT (
'RECNO  RATE  RANK  JOB   TASK' ),

        T3U               CHAR (29) STATIC INIT (
'------ ----- ---- ------ ----' ) ;

DECLARE WINDUPS (6)       CHAR (44) STATIC INIT (
             '    TOTAL NUMBER COURSE RECORDS PROCESSED : ',
             '   TOTAL NUMBER VEHICLE RECORDS PROCESSED : ',
             '   TOTAL NUMBER JOBTASK RECORDS PROCESSED : ',
             '     TOTAL NUMBER ROE RESULT RECORDS READ : ',
             'RECORDS WITHOUT MATCHING ABBREVIATED DATA : ',
             '   TOTAL NUMBER DIRECTORY SEARCH FAILURES : ' ),

        WIND (6)          FIXED BIN STATIC ;

/**************************************/

DECLARE 1 ACRS            STATIC,         /* ABBR. COURSE FILE */
          2 SET1,                         /* CKF.ACOURSES */
            3 CDP         CHAR (4),
            3 CIN         CHAR (8),
            3 CST         CHAR (16),
            3 NOBC        CHAR (4),
            3 NEC         CHAR (4),
          2 PC            CHAR (2),
          2 SET2,
            3 RMS         CHAR (3),
            3 TYCRS       CHAR (2),
            3 SVC         CHAR (1),
            3 MI          CHAR (1),
          2 STCD          CHAR (1),
          2 STDTE         PICTURE '(5)9',
          2 SET3,
            3 TRAPS       CHAR (1),
            3 TPC         CHAR (5),
          2 ATTR          PICTURE '99V9',
          2 STBK          PICTURE '99V9',
          2 CLEN          PICTURE '999',
          2 THRS          PICTURE '999',
          2 LHRS          PICTURE '999',
          2 CADB          PICTURE '(5)9V99',
          2 CCBO          CHAR (1) ;

DECLARE 1 CINDER (304) STATIC,            /* CRS CIN DIRECTORY */
          2 CHCIN         CHAR (8),       /* CKF.CRS.DIRCIN */
          2 CINPTR        FIXED BIN,

        LOADCIN           CHAR (760)  BASED (PLCIN),
        COVCIN (4)        CHAR (760)  BASED (PTCIN) ;

DECLARE 1 CDPDIR (1200) STATIC,           /* CRS CDP DIRECTORY */
          2 CHCDP         CHAR (4),       /* CKF.CRS.DIRCDP */
          2 CDPPTR        FIXED BIN,

        LOADCDP           CHAR (1200) BASED (PLCDP),
        COVCDP (6)        CHAR (1200) BASED (PTCDP) ;
```

```
PAGE NUMBER :    3

SOURCE CARD LISTING FOR
 P5A:     /* PRINT OF RAW ROE RESULTS FILE - MON/28/FEB/77 */


/****************************************/

DECLARE 1 AVEHICLES       STATIC,           /* ABBR. VEH FILE */
           2 STOCKN       CHAR (13),        /* CKF.AVEHS */
           2 DEVDESIG     CHAR (9),
           2 DEVNAME      CHAR (47),
           2 DEVCOST      PICTURE '(8)9V99',
           2 CC80         CHAR (1),

        VCOST             FIXED DEC (10,2) STATIC,
        VOVER             CHAR (80) BASED (PV80) ;

DECLARE 1 VEHDIR (200)    STATIC,           /* VEHICLE DIRECTORY */
           2 CHVEH        CHAR (13),        /* CKF.VEH.DIR */
           2 VEHPTR       FIXED BIN,

        COVVEH (4)        CHAR (750) BASED (PTVEH),
        LOADVEH           CHAR (750) BASED (PLVEH) ;

/****************************************/

DECLARE 1 ATASKS          STATIC,           /* ABBR. TASK FILE */
           2 RATING       CHAR (7),         /* CKF.ATASKS */
           2 JOBTASK      CHAR (6),
           2 TASKTTL      CHAR (50),
           2 FILL         CHAR (6),
           2 BILCOST      PICTURE '(8)9V99',
           2 CC80         CHAR (1) ;

DECLARE 1 TASKDIR (1200) STATIC,            /* JOBTASK FILE DIRECTORY */
           2 CHTASK       CHAR (13),        /* CKF.TASKS.DIR */
           2 TASKPTR      FIXED BIN,

        COVTASK (12)      CHAR (1500) BASED (PTTASK),
        LOADTASK          CHAR (1500) BASED (PLTASK) ;

/****************************************/

START:  NTYPE, NC, NJ, NV, NR, NN, NDF,
              NPAGE, DPAGE, LT,
              J, K = 0 ;

        FLCIN, FLCDP, FLVEH, FLTASK = '0'B ;

        PP = ADDR (INPARM) ;
        PR = ADDR (S16) ;
        PTCIN = ADDR (CINDER) ;
        PTCDP = ADDR (CDPDIR) ;
        PV80 = ADDR (AVEHICLES) ;
        PTVEH = ADDR (VEHDIR) ;
        PTTASK = ADDR (TASKDIR) ;

COMMENCE:
        OPEN FILE (MREF) RECORD INPUT,
             FILE (ID) RECORD INPUT,
             FILE (COURSE) INPUT,
             FILE (DCIN) RECORD INPUT,
             FILE (DCDP) RECORD INPUT,
             FILE (VEHS) INPUT,
             FILE (DVEH) RECORD INPUT,
             FILE (TASKS) INPUT,
             FILE (DTASK) RECORD INPUT,
             FILE (RESU) RECORD INPUT ;

        ON ENDFILE (RESU) GO TO EOF ;

        READ FILE (MREF) INTO (MAST) ;

 PAGE NUMBER :    4

 SOURCE CARD LISTING FOR
  P5A:    /* PRINT OF RAW ROE RESULTS FILE - MON/28/FEB/77 */
/*************************************/


   /* HERE, DEVICE ASSIGNMENTS ARE VERIFIED */

FIRST:  DISP, PTR = 1 ;
        IF ( PARM2 = 'BOTH' ) THEN GO TO STEPA ;

        IF ( PARM2 = 'TERM' )
             THEN PTR = 2 ;
             ELSE DISP = 2 ;

   /* HERE, PRINTER DATASET IS INITIALIZED */

STEPA:  IF ( PTR = 2 ) THEN GO TO STEPB ;
        OPEN FILE (SYSPRINT)
             LINESIZE (120) PAGESIZE (60) ;

        ON ENDPAGE (SYSPRINT)
             BEGIN ;
             NPAGE = NPAGE + 1 ;

             PUT EDIT ( 'PAGE NUMBER : ', NPAGE )
                  ( PAGE, X(10), A, F(4) ) ;

             IF ( NTYPE = 1 ) THEN
                  DO ;
                  PUT EDIT ( T1, T1U, '     ' ) ( R(FMT1) ) ;
                       GO TO SKP ;
                  END ;

FMT1:   FORMAT ( 3 ( SKIP(1), A ) ) ;

             IF ( NTYPE = 2 )
                  THEN PUT EDIT ( T2, T2U, '     ' ) ( R(FMT1) ) ;
                  ELSE PUT EDIT ( T3, T3U, '     ' ) ( R(FMT1) ) ;

SKP:         PUT SKIP (1) ;
             END ;

   /* HERE, INITILIZATION OF TERMINAL DATASET PROCEEDS */

STEPB:  IF ( DISP = 2 ) THEN GO TO STEPC ;

        ON CONDITION (NEWDP)
             BEGIN ;
             IF ( LT < 61 ) THEN LT = 61 ;

             IF ( LT < 66 ) THEN
LPA:              DO L = LT TO 66 ;
                  DISPLAY ( '    ' ) ;
                  END LPA ;

             DISPLAY ( '    ' ) ;

             DPAGE = DPAGE + 1 ;
             DDPGE = DPAGE ;
             DISPLAY ( DTTL ) ;
             DISPLAY ( '    ' ) ;
             LT = 3 ;

             DISPLAY ( DTTL2 ) ;  DISPLAY ( DTTL3 ) ;
             DISPLAY ( '    ' ) ;
             LT = LT + 3 ;
             END ;

   /* NOW, MASTER TITLE STRIP IS OUTPUT */

STEPC:  READ FILE (ID) INTO (CARD) ;
```

```
 PAGE NUMBER :   5

 SOURCE CARD LISTING FOR
  P5A:    /* PRINT OF RAW ROE RESULTS FILE - MON/28/FEB/77 */

        DLINE = (10)' ' || '** INITIAL SEARCH RESULTS FOR PROJECT -'
             || C110 || ' **' ;
        CARD = (12)' ' || C1180 ;
        DISPLAY ( '      ' ) ;   DISPLAY ( DLINE ) ;
        DISPLAY ( CARD ) ;   DISPLAY ( '      ' ) ;

        IF ( PTR = 0 ) THEN GO TO RD1 ;

        PUT EDIT ( DLINE, CARD ) ( SKIP(3), A, SKIP(1), A ) ;

             GO TO RD1 ;

/****************************************/

   /* HEREIN, IS START OF MAINLINE LOOP */

RD1:    READ FILE (RESU) INTO (S16) ;
        NR = NR + 1 ;

        IF ( RTYP = NTYPE ) THEN GO TO STEP1 ;

             NTYPE = RTYP ;
             D2TY = VINTAGE (NTYPE) ;
             IF ( PTR = 1 ) THEN SIGNAL ENDPAGE (SYSPRINT) ;
             IF ( DISP = 1 ) THEN SIGNAL CONDITION (NEWDP) ;

STEP1:  GO TO V1 (NTYPE) ;

   /* HERE, A COURSE RECORD IS PROCESSED */

V1(1):  NC = NC + 1 ;

        IF ( RDSTAT = 2 ) THEN
NOCDATA:     DO ;
             NN = NN + 1 ;

             PUT STRING (DLINE) EDIT
                  ( NR, RESCIN, 'NOTE - NO MATCHING NITRAS DATA',
                    ' AVAILABLE FOR THIS CIN' )
                  ( P'ZZ,ZZ9', X(1), A(8), X(2), A, A ) ;

                       GO TO OUTLOOP ;
             END NOCDATA ;

        YECDP = RESCDP ;
        LOOKU = STEP2 ;
             GO TO FINDCDP ;

STEP2:  IF ( CDPREC = 0 ) THEN
LP1:         DO ;
             NDF = NDF + 1 ;

             PUT STRING (DLINE) EDIT
                  ( NR, RESCIN, RESCDP, '** CDP COULD NOT ',
                  'BE LOCATED IN CDP DIRECTORY' )
                  ( P'ZZ,ZZ9', X(1), A(8), X(1), A(4),
                    X(2), A, A ) ;

                       GO TO OUTLOOP ;
             END LP1 ;

        READ FILE (COURSE) INTO (ACRS) KEY (CDPREC) ;

        PUT STRING (DLINE) EDIT ( NR, RESCIN, RESCUP, CST )
             ( P'ZZ,ZZ9', X(1), A(8), X(1), A(4), X(1), A(16) ) ;

                  GO TO OUTLOOP ;
 PAGE NUMBER :   6

 SOURCE CARD LISTING FOR
  P5A:     /* PRINT OF RAW RDE RESULTS FILE - MON/28/FEB/77 */

   /* HERE, VEHICLE TYPE RECORDS ARE PROCESSED */

V1(2):  NV = NV + 1 ;

        IF ( RDSTAT = 2 ) THEN
NOVDATA:     DO ;
             NN = NN + 1 ;

             PUT STRING (DLINE) EDIT
                  ( NR, RESID, '** NO ABBREVIATED VEHICLE DATA ',
                    'AVAILABLE **' )
                  ( P'ZZ,ZZ9', X(1), A(13), X(2), A, A ) ;

                       GO TO OUTLOOP ;
             END NOVDATA ;

        YEVEH = RESID ;
        LOOKU = STEP3 ;
             GO TO FINDVEH ;

STEP3:  PUT STRING (DLINE) EDIT ( NR, RESID )
             ( P'ZZ,ZZ9', X(1), A(13) ) ;

        IF ( VEHREC = 0 ) THEN
LP2:         DO ;
             NDF = NDF + 1 ;

             SUBSTR ( DLINE, 22 ) =
                  '** STOCK NUMBER COULD NOT BE LOCATED IN ' ||
                  'DIRECTORY **' ;

                       GO TO OUTLOOP ;
             END LP2 ;

        READ FILE (VEHS) INTO (AVEHICLES) KEY (VEHREC) ;

        SUBSTR ( DLINE, 22 ) =
             DEVDESIG || ' ' || DEVNAME ;

                  GO TO OUTLOOP ;

   /* NOW, PROCESS A TASK RECORD HERE */

V1(3):  NJ = NJ + 1 ;

        PUT STRING (DLINE) EDIT ( NR, RATE, RANK, REST.JOBTASK )
             ( P'ZZ,ZZ9', X(1), A(5), X(1), A(2), X(1), A(6) ) ;

        IF ( RDSTAT = 2 ) THEN
NOTDATA:     DO ;
             NN = NN + 1 ;

             SUBSTR ( DLINE, 26 ) =
                  '** NO ABBREVIATED TASK DATA IS AVAILABLE **' ;

                       GO TO OUTLOOP ;
             END NOTDATA ;

        YETASK = RESID ;
        LOOKU = STEP4 ;
             GO TO FINDTASK ;

STEP4:  IF ( TASKREC = 0 ) THEN
LP3:         DO ;
             NDF = NDF + 1 ;

             SUBSTR ( DLINE, 26 ) =
                  '** TASK ID COULD NOT BE LOCATED IN DIRECTORY **' ;
 PAGE NUMBER :    7

 SOURCE CARD LISTING FOR
  P5A:     /* PRINT OF RAW ROE RESULTS FILE - MON/28/FEB/77 */
                       GO TO OUTLOOP ;
             END LP3 ;

        READ FILE (TASKS) INTO (ATASKS) KEY (TASKREC) ;

        SUBSTR ( DLINE, 26 ) = TASKTTL ;

             GO TO OUTLOOP ;

   /* HERE IS THE PRINTED LINE OUTPUT SEQUENCE */

OUTLOOP: IF ( PTR = 1 )
             THEN PUT LIST (DLINE) SKIP (1) ;

        IF ( DISP = 2 ) THEN GO TO RD1 ;

        IF ( LT > MAXLTB ) THEN SIGNAL CONDITION (NEWDP) ;
        DISPLAY ( DLINE ) ;
        LT = LT + 1 ;

             GO TO RD1 ;

/*****************************************/

DECLARE ( FLCIN, FLCDP, FLVEH, FLTASK )    BIT (1) STATIC,
        YECIN          CHAR (8) STATIC,
        YECDP          CHAR (4) STATIC,
        ( YETASK, YEVEH )   CHAR (13) STATIC,
        ( CINREC, CDPREC, VEHREC, TASKREC, VEHMAX, TASKMAX,
             CINMAX, CDPMAX )    FIXED BIN STATIC,
        LOOKU          LABEL ;

   /* CIN DIRECTORY LOOKUP SUBROUTINE

             ENTRY : DESIRED CIN IN YECIN
             EXIT  : CINREC = 0 (NO FIND) OR REG NUMBER
             RETURN VECTOR IS LOOKU          */

FINDCIN: IF ( FLCIN ) THEN GO TO CIN2 ;

        CINMAX = REFS (2) ;
        FLCIN = '1'B ;

CINLOAD:     DO J = 1 TO 4 ;
             PLCIN = ADDR ( COVCIN (J) ) ;
             READ FILE (DCIN) INTO (LOADCIN) ;
             END CINLOAD ;

CIN2:        DO I = 1 TO CINMAX ;
             IF ( YECIN ¬= CHCIN(I) ) THEN GO TO ECIN2 ;
                  CINREC = CINPTR (I) ;
                       GO TO LOOKU ;
ECIN2:       END CIN2 ;

NFCIN:  CINREC = 0 ;
             GO TO LOOKU ;

   /* CDP DIRECTORY LOOKUP SUBROUTINE
             ENTRY : DESIRED CDP IN YECDP
             EXIT  : CDPREC = 0 (NO FIND), OR REC NUMBER
           . RETURN VECTOR IS LOOKU          */

FINDCDP: IF ( FLCDP ) THEN GO TO CDP2 ;

        CDPMAX = REFS (3) ;
        FLCDP = '1'B ;

CDPLOAD:     DO J = 1 TO 4 ;
             PLCDP = ADDR ( COVCDP (J) ) ;
```

```
PAGE NUMBER :    8

SOURCE CARD LISTING FOR
 P5A:     /* PRINT OF RAW ROE RESULTS FILE - MON/28/FEB/77 */

                   READ FILE (DCDP) INTO (LOADCDP) ;
                   END CDPLOAD ;

CDP2:              DO I = 1 TO CDPMAX ;
                   IF ( YECDP ¬= CHCDP(I) ) THEN GO TO ECDP2 ;
                          CDPREC = CDPPTR(I) ;
                                 GO TO LOOKU ;
ECDP2:             END CDP2 ;

NFCDP:  CDPREC = 0 ;
                   GO TO LOOKU ;

    /* VEHICLE ID LOOKUP SUBROUTINE
                   ENTRY : DESIRED VEH ID IN YEVEH
                   EXIT  : VEHREC = 0 (NO FIND), OR REC NUMBER
                   RETURN VECTOR IS LOOKU              */

FINDVEH: IF ( FLVEH ) THEN GO TO VEH2 ;

             VEHMAX = REFS (8) ;
             FLVEH = '1'B ;

VEHLOAD:           DO J = 1 TO 4 ;
                   PLVEH = ADDR ( COVVEH (J) ) ;
                   READ FILE (DVEH) INTO (LOADVEH) ;
                   END VEHLOAD ;

VEH2:              DO I = 1 TO VEHMAX ;
                   IF ( YEVEH ¬= CHVEH(I) ) THEN GO TO EVEH2 ;
                          VEHREC = VEHPTR (I) ;
                                 GO TO LOOKU ;
EVEH2:             END VEH2 ;

NFVEH:  VEHREC = 0 ;
                   GO TO LOOKU ;

    /* TASK ID DIRECTORY LOOKUP SUBROUTINE
                   ENTRY : DESIRED TASK ID (13-CHAR) IN YETASK
                   EXIT  : TASKREC = 0 (NO FIND) OR REC NUMBER
                   RETURN VECTOR IS LOOKU              */

FINDTASK: IF ( FLTASK ) THEN GO TO TASK2 ;

                TASKMAX = REFS (12) ;
             FLTASK = '1'B ;

TASKLOAD:          DO J = 1 TO 12 ;
                   PLTASK = ADDR ( COVTASK (J) ) ;
                   READ FILE (DTASK) INTO (LOADTASK) ;
                   END TASKLOAD ;

TASK2:             DO I = 1 TO TASKMAX ;
                   IF ( YETASK ¬= CHTASK(I) ) THEN GO TO ETASK2 ;
                          TASKREC = TASKPTR (I) ;
                                 GO TO LOOKU ;
ETASK2:            END TASK2 ;

NFTASK: TASKREC = 0 ;
                   GO TO LOOKU ;

/*****************************************/

    /* HERE IS EOF WINDUP PROCESSING */

EOF:        WIND(1) = NC ;        WIND(2) = NV ;
            WIND(3) = NJ ;        WIND(4) = NR ;
            WIND(5) = NN ;        WIND(6) = NDF ;

            IF ( PTR = 1 ) THEN PUT SKIP (3) ;
```

```
PAGE NUMBER :    9

SOURCE CARD LISTING FOR
 P5A:      /* PRINT OF RAW ROE RESULTS FILE - MON/28/FEB/77 */


               DISPLAY ( '     ' ) ;   DISPLAY ( '     ' ) ;
               DISPLAY ( '** RESULTS FILE NOW AT END-OF-FILE **' ) ;
               DISPLAY ( '     ' ) ;

SUMMARY:            DO J = 1 TO 6 ;

                    PUT STRING (DLINE) EDIT
                         ( WINDUPS(J), WIND(J) ) ( A(44), P'ZZ,ZZ9' ) ;

                    IF ( PTR = 1 )
                         THEN PUT LIST ( (10)' ' || DLINE ) SKIP (1) ;

                    DISPLAY ( DLINE ) ;

                    END SUMMARY ;

               CLOSE FILE (MREF), FILE (ID), FILE (RESU),
                     FILE (COURSE), FILE (DCDP),
                     FILE (VEHS), FILE (DVEH),
                     FILE (TASKS), FILE (DTASK) ;

               IF ( PTR = 1 ) THEN CLOSE FILE (SYSPRINT) ;
               DISPLAY ( '     ' ) ;

                    END P5A ;
 PAGE NUMBER :    1

 SOURCE CARD LISTING FOR
  P5B:     /* INTERACTIVE SEARCH RESULTS EDITOR - WED/09/MAR/77 */

P5B:      /* INTERACTIVE SEARCH RESULTS EDITOR - WED/09/MAR/77 */
          PROC OPTIONS (MAIN) ;

   /*     REVISIONS :
          FRI/11/MAR/77 - VERYFIRST IMPLEMENTATION.

          MON/21/MAR/77 - MODS PRIOR TO TRANSMIT TO NCSS.
   */

DECLARE ( TOTREC, MOD,  N, N1, N2, L1, L2, L3, NREC, NRECA, NADD,
               NLINE, NEE, I, J1, J2, J3, J4, MAXADD, MAXD, MAXR,
               NCONV, NER, NFIRST, NN, NOS,
               J, K, L ) FIXED BIN STATIC,

          ( CARD, DLINE ) CHAR (80) STATIC,
          C110                CHAR (10) DEF CARD POS (1),
          C1180               CHAR (70) DEF CARD POS (11),
          C15                 CHAR (5) DEF CARD,
          C16                 CHAR (6) DEF CARD,
          C12                 CHAR (2) DEF CARD,
          C17                 CHAR (7) DEF CARD,
          CTAB (80)           CHAR (1) DEF CARD,

          ( N1C, N2C )    CHAR (5) STATIC,
          ( ROUTEX, V1(3),
               ROUTE )    LABEL ;

DECLARE ( COURSE, VEHS,
               TASKS )    FILE RECORD DIRECT KEYED ENV ( REGIONAL(1) ) ;

DECLARE ADDS (500)        CHAR (16) STATIC,

          NEW                 CHAR (16) STATIC,
          NTYP                PICTURE '9' DEF NEW POS (1),
          NSTAT               PICTURE '9' DEF NEW POS (2),
          NDSTAT              PICTURE '9' DEF NEW POS (3),
          NID                 CHAR (13) DEF NEW POS (4),
          NCIN                CHAR (8)  DEF NEW POS (4),
          NCDP                CHAR (4)  DEF NEW POS (12),
          NRATE               CHAR (5)  DEF NEW POS (4),
          NRANK               CHAR (2)  DEF NEW POS (9),
          NJTASK              CHAR (6)  DEF NEW POS (11),

          1 MODS (500)    BASED (PM),
            2 MTYP        CHAR (1),
            2 MSTAT       CHAR (1),
            2 MDSTAT      CHAR (1),
            2 FILL        CHAR (13) ;

DECLARE 1 RESULT (1200) STATIC,
            2 RTYP        PICTURE '9',
            2 RSTAT       PICTURE '9',
            2 RDSTAT      PICTURE '9',
            2 RESID       CHAR (13),

          RESREC (1200)   CHAR (16) BASED (PR),
          S16             CHAR (16) STATIC,

          1 RESC (1200)   BASED (PR),
            ( 2 F1, 2 F2, 2 F3 )   CHAR (1),
            2 RESCIN      CHAR (8),
            2 RESCDP      CHAR (4),
            2 F4          CHAR (1),

          1 REST (1200)   BASED (PR),
            ( 2 F1, 2 F2, 2 F3 )   CHAR (1),
            2 RATE        CHAR (5),
            2 RANK        CHAR (2),
PAGE NUMBER :    2

SOURCE CARD LISTING FOR
 P5B:     /* INTERACTIVE SEARCH RESULTS EDITOR - WED/09/MAR/77 */
            2 JOBTASK     CHAR (6) ;

DECLARE 1 MAST            STATIC,
            2 REFS (40)   FIXED BIN ;

DECLARE 1 TTLCARD         STATIC,            /* PROJECT TITLE DATA BASE */
            2 PNAME       CHAR (10),         /* CKF.TITLE */
            2 PTTL        CHAR (70),

          INPARM          CHAR (100) VARYING,

          1 POVER         BASED (PP),
            2 PARM1       CHAR (4),
            2 PARM2       CHAR (4) ;

/****************************************/

DECLARE 1 CINDER (304) STATIC,               /* CRS CIN DIRECTORY */
            2 CHCIN       CHAR (8),          /* CKF.CRS.DIRCIN */
            2 CINPTR      FIXED BIN,

          LOADCIN         CHAR (760)  BASED (PLCIN),
          COVCIN (4)      CHAR (760)  BASED (PTCIN) ;

DECLARE 1 CDPDIR (1200) STATIC,              /* CRS CDP DIRECTORY */
            2 CHCDP       CHAR (4),          /* CKF.CRS.DIRCDP */
            2 CDPPTR      FIXED BIN,

          LOADCDP         CHAR (1200) BASED (PLCDP),
          COVCDP (6)      CHAR (1200) BASED (PTCDP) ;

DECLARE 1 ACRS            STATIC,            /* ABBR. COURSE FILE */
            2 SET1,                          /* CKF.ACOURSES */
              3 CDP       CHAR (4),
              3 CIN       CHAR (8),
              3 CST       CHAR (16),
              3 NOBC      CHAR (4),
              3 NEC       CHAR (4),
            2 PC          CHAR (2),
            2 SET2,
              3 RMS       CHAR (3),
              3 TYCRS     CHAR (2),
              3 SVC       CHAR (1),
              3 MI        CHAR (1),
            2 STCD        CHAR (1),
            2 STDTE       PICTURE '(5)9',
            2 SET3,
              3 TRAPS     CHAR (1),
              3 TPC       CHAR (5),
            2 ATTR        PICTURE '99V9',
            2 STBK        PICTURE '99V9',
            2 CLEN        PICTURE '999',
            2 THRS        PICTURE '999',
            2 LHRS        PICTURE '999',
            2 CADB        PICTURE '(5)9V99',
            2 CC80        CHAR (1) ;

/****************************************/

DECLARE 1 VEHDIR (200)  STATIC,              /* VEHICLE DIRECTORY */
            2 CHVEH       CHAR (13),         /* CKF.VEH.DIR */
            2 VEHPTR      FIXED BIN,

          COVVEH (4)      CHAR (750) BASED (PTVEH),
          LOADVEH         CHAR (750) BASED (PLVEH) ;

DECLARE 1 AVEHICLES       STATIC,            /* ABBR. VEH FILE */
            2 STOCKN      CHAR (13),         /* CKF.AVEHS */
            2 DEVDESIG    CHAR (9),
 PAGE NUMBER :    3

 SOURCE CARD LISTING FOR
  P5B:     /* INTERACTIVE SEARCH RESULTS EDITOR - WED/09/MAR/77 */

                2 DEVNAME     CHAR (47),
                2 DEVCOST     PICTURE '(8)9V99',
                2 CC80        CHAR (1),

            VOVER             CHAR (80) BASED (PV80) ;

/****************************************/

DECLARE 1 TASKDIR (1200) STATIC,          /* JOBTASK FILE DIRECTORY */
            2 CHTASK          CHAR (13),  /* CKF.TASKS.DIR */
            2 TASKPTR         FIXED BIN,

          COVTASK (12)    CHAR (1500) BASED (PTTASK),
          LOADTASK        CHAR (1500) BASED (PLTASK) ;

DECLARE 1 ATASKS          STATIC,         /* ABBR. TASK FILE */
            2 RATING      CHAR (7),       /* CKF.ATASKS */
            2 JOBTASK     CHAR (6),
            2 TASKTTL     CHAR (50),
            2 FILL        CHAR (6),
            2 BILCOST     PICTURE '(8)9V99',
            2 CC80        CHAR (1) ;

/****************************************/

START:  MOD,  TOTREC, NADD, MAXADD, NEE,
              J, K = 0 ;

          PM = ADDR (ADDS) ;
          PR = ADDR (RESULT) ;
          PTCIN = ADDR (CINDER) ;
          PTCDP = ADDR (CDPDIR) ;
          PTVEH = ADDR (VEHDIR) ;
          PV80 = ADDR (AVEHICLES) ;
          PTTASK = ADDR (TASKDIR) ;
          PP = ADDR (INPARM) ;

          ADDS = ' ' ;

COMMENCE:
          OPEN FILE (MREF) RECORD INPUT,
               FILE (ID) RECORD INPUT,
               FILE (COURSE) INPUT,
               FILE (DCIN) RECORD INPUT,
               FILE (DCDP) RECORD INPUT,
               FILE (VEHS) INPUT,
               FILE (DVEH) RECORD INPUT,
               FILE (TASKS) INPUT,
               FILE (DTASK) RECORD INPUT,
               FILE (RESU) RECORD INPUT;

          READ FILE (MREF) INTO (MAST) ;

          ON CONVERSION
               BEGIN ;
               DLINE = '      ** ILLEGAL NUMERIC CONVERSION **' ;
          DISPLAY ( '     ' ) ;  DISPLAY ( DLINE ) ;
                    GO TO PROMPTA ;
               END ;

    /* HERE, LEADIN TITLE STRIP IS PRODUCED */

          READ FILE (ID) INTO (CARD) ;

          DLINE = (10)' ' || '** ROE SEARCH RESULTS EDIT FOR PROJECT '
               || C110 || ' **' ;
          CARD = (13)' ' || C1180 ;

          DISPLAY ( '     ' ) ;  DISPLAY ( DLINE ) ;
```

```
PAGE NUMBER :   4

SOURCE CARD LISTING FOR
 P58:     /* INTERACTIVE SEARCH RESULTS EDITOR - WED/09/MAR/77 */
          DISPLAY ( CARD ) ;
          DISPLAY ( '     ' ) ;

/****************************************/

   /* FIRST, FETCH THE RESULTS FILE INTO MS */

          ON ENDFILE (RESU) GO TO LOAD2 ;

LOADR:           DO J = 1 TO 1200 ;
                 READ FILE (RESU) INTO (S16) ;
                 TOTREC = J ;
                 RESREC (J) = S16 ;
                 END LOADR ;

ER1:      NER = 1 ;
          DLINE = '    MORE THAN 1200 RESULTS RECORDS - CANNOT EDIT' ;
                 GO TO BUSTED ;

LOAD2:    CLOSE FILE (RESU) ;

          PUT STRING (DLINE) EDIT
                 ( '** ', TOTREC, ' HAVE BEEN LOADED INTO ',
                   'MEMORY FOR EDITING' ) ( A, F(4), A, A ) ;

          DISPLAY ( '     ' ) ;  DISPLAY ( DLINE ) ;

/****************************************/

   /* HERE IS CENTRAL PROMPT LOOP AND REPLY-LOOKUP SEQUENCE */

PROMPTA:    DISPLAY ( '     ' ) ;
          DISPLAY    ( '** MAKE NEXT ACTION REQUEST' )  REPLY ( CARD ) ;

          IF ( CARD = ' ' ) THEN GO TO PROMPTA ;

          IF ( CTAB(1) ¬= ' ' ) THEN GO TO ELOOK ;
LP1:             DO J = 1 TO 20 ;
                 IF ( CTAB(J) = ' ' ) THEN GO TO ELP1 ;
                      CARD = SUBSTR ( CARD, J ) ;
                           GO TO ELOOK ;
ELP1:            END LP1 ;

BUMENTRY: DISPLAY ( '     ' ) ;
            DISPLAY  ( (10)' ' || '** CANNOT DECODE LAST ENTRY '
                 || '- TRY AGAIN' ) ;
                      GO TO PROMPTA ;

ELOOK:    IF ( C12 = 'D ' ) | ( C17 = 'DELETE ' )
                 THEN GO TO DELETE ;

          IF ( C12 = 'P ' ) | ( C16 = 'PRINT ' )
                 THEN GO TO PRINT ;

          IF ( C15 = 'ADDC ' ) THEN GO TO ADDCOURSE ;
          IF ( C15 = 'ADDV ' ) THEN GO TO ADDVEHICLE ;
          IF ( C15 = 'ADDJ ' ) | ( C15 = 'ADDT ' )
                 THEN GO TO ADDJOBTASK ;
          IF ( C15 = 'QUIT ' ) THEN GO TO QUIT ;

                      GO TO BUMENTRY ;

/****************************************/

   /* THIS IS PROCESSOR FOR SINGLE / MULTIPLE DELETES */

DELETE: ROUTE = STEP1 ;
                 GO TO FINDNOS ;
 PAGE NUMBER :   5

 SOURCE CARD LISTING FOR
  P5B:     /* INTERACTIVE SEARCH RESULTS EDITOR - WED/09/MAR/77 */

STEP1:  K = N1 ;
        IF ( N2 = 0 ) THEN N2 = N1 ;

STEP2:        ROUTE = STEP3 ;
                    GO TO FINDREC ;

STEP3:  IF ( K > TOTREC )
              THEN DO ;  MTYP (NRECA) = '0' ;  END ;
              ELSE DO ;  RDSTAT (NREC) = 3 ;  END ;

        K = K + 1 ;
        IF ( K <= N2 ) THEN GO TO STEP2 ;

        DLINE = 'DELETED RECORD NUMBER : ' || N1C ;
        IF ( NOS = 2 ) THEN
              SUBSTR (DLINE,31) = 'THRU ' || N2C ;

STEP4:  DISPLAY ( DLINE ) ;

STEP5:  MOD = 1 ;               /* SIGNAL, FILE HAS BEEN CHANGED */

              GO TO PROMPTA ;

/****************************************/

   /* HEREIN IS THE PROCESSING FOR THE KEYBOARD PRINT COMMAND */

PRINT:  ROUTE = STEP6 ;
              GO TO FINDNOS ;

STEP6:  K = N1 ;
        IF ( NOS = 1 ) THEN N2 = N1 ;

STEP7:  ROUTE = STEP8 ;
              GO TO FINDREC ;

STEP8:  IF ( K > TOTREC )
              THEN NEW = ADDS (NRECA) ;
              ELSE NEW = RESREC (NREC) ;

        N = NTYP ;
        IF ( N > 0 ) & ( N < 4 ) THEN GO TO V1 (N) ;

ER2:          NER = 2 ;
              PUT STRING (DLINE) DATA ( NEW, N, K ) ;
                    GO TO BUSTED ;

   /* HERE, COURSE RESULTS ARE PRINTED */

V1(1):  MAXD = REFS (2) ;    /* MAX SS USED IN DIRECTORY */
        MAXR = REFS (1) ;    /* MAX COURSE REC NUM */
        NLINE = 1 ;          /* ALWAYS 1 LINE WITH COURSES */

        PUT STRING (DLINE) EDIT ( 'RECNO/CIN/CDP/TITLE : ',
              K, '/', NCIN, '/', NCDP, '/ ' )
              ( A, P'ZZ,ZZ9', A, A(8), A, A(4), A ) ;

        IF ( NOSTAT = 1 ) THEN GO TO STEP9 ;
        SUBSTR (DLINE,45) = '(NO MATCHING DATA)' ;
              GO TO OUTLOOP ;

STEP9:  YECDP = NCDP ;
        LOOKU = STEP10 ;
              GO TO FINDCDP ;

        IF ( CDPREC > 0 ) THEN GO TO STEP10 ;
        SUBSTR (DLINE,45) = 'CDP NOT IN DIRECTORY' ;
              GO TO OUTLOOP ;
 PAGE NUMBER :    6

 SOURCE CARD LISTING FOR
  P5B:     /* INTERACTIVE SEARCH RESULTS EDITOR - WED/09/MAR/77 */
STEP10: IF ( CDPREC <= MAXR ) THEN GO TO STEP11 ;
ER3:             NER = 3 ;
                 PUT STRING (DLINE) DATA ( CDPREC, MAXR, K ) ;
                      GO TO BUSTED ;

STEP11: READ FILE (COURSE) INTO (ACRS) KEY (CDPREC) ;

        SUBSTR (DLINE,45) = CST ;

             GO TO OUTLOOP ;

   /* HERE, VEHICLE RECORD PRINTING IS SETUP */

V1(2):  MAXD = REFS (8) ;
        MAXR = REFS (7) ;
        NLINE = 1 ;
        NN = 46 ;

        PUT STRING (DLINE) EDIT ( 'RECNO/STOCK/FILE NUM : ', K,
            '/', NID, '/' ) ( A, P'ZZ,ZZ9', A, A(13), A ) ;

        IF ( NDSTAT = 1 ) THEN GO TO STEP12 ;
STP12A: SUBSTR (DLINE,NN) = '(NO MATCHING DATA)' ;
             GO TO OUTLOOP ;

STEP12: YEVEH = NID ;
        LOOKU = STEP13 ;
             GO TO FINDVEH ;

STEP13: IF ( VEHREC > 0 ) THEN GO TO STEP14 ;
        SUBSTR (DLINE,45) = 'VEHICLE NOT IN DIRECTORY' ;
             GO TO OUTLOOP ;

STEP14: IF ( VEHREC <= MAXR ) THEN GO TO STEP15 ;
ER4:             NER = 4 ;
                 PUT STRING (DLINE) DATA ( VEHREC, MAXR, K ) ;
                      GO TO BUSTED ;

STEP15: READ FILE (VEHS) INTO (AVEHICLES) KEY (VEHREC) ;

        SUBSTR (DLINE,45) = DEVDESIG ;
        CARD = ' ' ;
        SUBSTR (CARD,21) = DEVNAME ;

STP15A:          NLINE = 2 ;
                      GO TO OUTLOOP ;

   /* HERE, FINALLY, JOBTASK PROCESSING ENSUES */

V1(3):  MAXD = REFS (12) ;
        MAXR = REFS (11) ;
        NLINE = 1 ;
        NN = 47 ;

        PUT STRING (DLINE) EDIT ( 'RECNO/RATE/RANK/JOB  : ', K,
            '/', NRATE, '/', NRANK, '/', NJTASK )
            ( A, P'ZZ,ZZ9', A, A, A, A, A, A ) ;

        IF ( NDSTAT = 2 ) THEN GO TO STP12A ;

        YETASK = NID ;
        LOOKU = STEP16 ;
             GO TO FINDTASK ;

STEP16: IF ( TASKREC > 0 ) THEN GO TO STEP17 ;
        SUBSTR (DLINE,47) = 'JOBTASK NOT IN DIRECTORY' ;
             GO TO OUTLOOP ;

STEP17: IF ( TASKREC <= MAXR ) THEN GO TO STEP18 ;
```

```
 PAGE NUMBER :    7

 SOURCE CARD LISTING FOR
  P5B:     /* INTERACTIVE SEARCH RESULTS EDITOR - WED/09/MAR/77 */

ER5:            NER = 5 ;
                PUT STRING (DLINE) DATA ( TASKREC, MAXR, K ) ;
                     GO TO BUSTED ;

STEP18: READ FILE (TASKS) INTO (ATASKS) KEY (TASKREC) ;

        CARD = ' ' ;
        SUBSTR ( CARD, 10 ) = TASKTTL ;

                GJ TO STP15A ;

   /* HERE IS THE COMMON OUTPUT SFQUENCE FOR COURSES, VEHICLES,
      AND TASKS RECORD PRINT */

OUTLOOP: DISPLAY ( '     ' ) ;
        DISPLAY ( DLINE ) ;

         IF ( NLINE = 2 ) THEN
                DISPLAY ( CARD ) ;

        K = K + 1 ;
         IF ( K <= N2 ) THEN GO TO STEP7 ;

        DISPLAY ( '     ' ) ;
                GO TO PROMPTA ;

/****************************************/

   /* HERE WE HAVE THE PROCESSING ROUTINES FOR THE ADD-COURSES
      (ADDC) KEYBOARD ENTRY */

ADDCOURSE:
ADDC:   IF ( MAXADD > 0 ) THEN GO TO BUMADD ;

        MAXD = REFS (2) ;
        MAXR = REFS (1) ;
        NFIRST = 1 ;
        NLINE = 1 ;

        ROUTEX = PROMPTA ;
        ROUTE = STEP19 ;
                GJ TO FINDASLASH ;

STEP19: YECIN = SUBSTR ( CARD, (J1+1), (J2-J1-1) ) ;
        YECDP = ' ' ;

        IF ( J3 = 0 ) | ( (J3 - J2) = 1 ) THEN GO TO STP19A ;
        YECDP = SUBSTR ( CARD, (J2 + 1), (J3 - J2 - 1) ) ;

STP19A: NTYP = 1 ;  NSTAT = 2 ;  NOSTAT = 2 ;
        NID = ' ' ;  NCIN = YECIN ;  NCDP = YECDP ;

        NADD = NADD + 1 ;
        IF ( NADD <= 500 ) THEN GO TO STEP20 ;

BUMADD:         NADD = NADD - 1 :
                MAXADD = NADD ;

                DLINE = '** A TOTAL OF ' || NADD || ' NEW RECORDS HAVE '
                     || 'BEEN ADDED' ;
        DISPLAY ( '     ' ) ;  DISPLAY ( DLINE ) ;

                DLINE = '    ANOTHER RUN OF THIS ROUTINE MUST BE MADE '
                     || 'TO ENTER FURTHER ADDITIONS' ;
        DISPLAY ( DLINE ) ;

                     GO TO PROMPTA ;

STEP20: NREC = TOTREC + NADD ;
```

```
        DLINE = 'THE FOLLOWING RECORD(S) HAS/HAVE BEEN ADDED...' ;
        DISPLAY ( '     ' ) ;  DISPLAY ( DLINE ) ;

        PUT STRING (DLINE) EDIT ( (10)' ', 'RECNO/CIN/CDP/COURSE : ',
             NREC, '/', YECIN, '/', YECDP, '/' )
             ( A, A, P'ZZ,ZZ9', A, A(8), A, A(4), A ) ;

        IF ( YECDP = ' ' ) THEN GO TO STEP23 ;

   /* HERE, THE ENTERED CDP WAS NOT BLANK, LOOK IT UP */

        LOOKU = STEP21 ;
             GO TO FINDCDP ;

STEP21: IF ( CDPREC > 0 ) THEN GO TO STEP22 ;
STEP21A:     SUBSTR (DLINE, 55) = '(NO MATCHING DATA)' ;
                  GO TO ADDOUT ;

STEP22: IF ( CDPREC > MAXR ) THEN GO TO ER3 ;

        READ FILE (COURSE) INTO (ACRS) KEY (CDPREC) ;

        SUBSTR (DLINE, 55) = CST ;
        NDSTAT = 1 ;

                  GO TO ADDOUT ;

   /* HERE, THE GIVEN CDP WAS BLANK ; TRY CIN LOOKUP */

STEP23: LOOKU = STEP24 ;
             GO TO FINDCIN ;

STEP24: IF ( CINREC = 0 ) THEN GO TO STEP21A ;

             ROUTEX = STEP26 ;

        IF ( CINREC > MAXR ) THEN GO TO ER3 ;

STEP25: READ FILE (COURSE) INTO (ACRS) KEY (CINREC) ;

        IF ( CIN ¬= YECIN ) THEN GO TO PROMPTA ;

        SUBSTR ( DLINE, 49 ) = CDP || '/ ' || CST ;
        YECDP = CDP ;  NCDP = CDP ;

        IF ( NFIRST = 1 ) THEN
             DO ;  NFIRST = 2 ;  GO TO ADDOUT ;  END ;

        NADD = NADD + 1 ;
        IF ( NADD > 500 ) THEN GO TO BUMADD ;
        NREC = TOTREC + NADD ;

        PUT STRING (SUBSTR(DLINE,33,6)) EDIT (NREC) ( P'ZZ,ZZ9' ) ;

   /* HERE IS THE COMMON OUTPUT ROUTINE FOR ADDITIONS */

ADDOUT: DISPLAY ( '     ' ) ;
        DISPLAY ( DLINE ) ;

        IF ( NLINE = 2 ) THEN
             DISPLAY ( CARD ) ;

        ADDS (NADD) = NEW ;

             GO TO ROUTEX ;  /* NORMAL ROUTE IS TO PROMPTX */

STEP26: /* HERE, FROM ROUTEX, WHEN POSSIBLY ANOTHER CIN/CDP
           RECORD TO BE EXTRACTED */
        CINREC = CINREC + 1 ;
        IF ( CINREC > MAXR ) THEN GO TO PROMPTA ;
                             ELSE GO TO STEP25 ;

/***************************************/

   /* NOW, THE PROCESSING ENSUES FOR ADDITIONS OF VEHICLE RECORDS */

ADDVEHICLE:
ADDV:   IF ( MAXADD > 0 ) THEN GO TO BUMADD ;

        ROUTEX = PROMPTA ;
        NLINE = 1 ;
        MAXD = REFS (8) ;
        MAXR = REFS (7) ;

        ROUTE = STEP27 ;
             GO TO FINDASLASH ;

STEP27: YEVEH = SUBSTR ( CARD, (J1 + 1), (J2 - J1 - 1) ) ;
        NTYP = 2 ;  NSTAT = 2 ;  NDSTAT = 2 ;
        NID = YEVEH ;

        NADD = NADD + 1 ;
        IF ( NADD > 500 ) THEN GO TO BUMADD ;
        NREC = TOTREC + NADD ;

        DLINE = 'THE FOLLOWING RECORD HAS BEEN ADDED...' ;
        DISPLAY ( '    ' ) ;  DISPLAY ( DLINE ) ;

        PUT STRING (DLINE) EDIT ( (10)' ', 'RECNO/STOCK/FILE NUM : ',
             NREC, '/', YEVEH, '/' ) ( A, A, P'ZZ,ZZ9', A, A(13), A ) ;

        LOOKU = STEP28 ;
             GO TO FINDVEH ;

STEP28: IF ( VEHREC > 0 ) THEN GO TO STEP29 ;
             SUBSTR ( DLINE, 56 ) = '(NO MATCHING DATA)' ;
                  GO TO ADDOUT ;

STEP29: IF ( VEHREC > MAXR ) THEN GO TO ER4 ;

        READ FILE (VEHS) INTO (AVEHICLES) KEY (VEHREC) ;

        SUBSTR ( DLINE, 55 ) = DEVDESIG ;

        CARD = ' ' ;
        SUBSTR ( CARD, 21 ) = DEVNAME ;
        NDSTAT = 1 ;
        NLINE = 2 ;

             GO TO ADDOUT ;

/***************************************/

   /* FINALLY, THE TASK ADDITION KEYBOARD ENTRY IS PURSUED */

ADDJOBTASK:
ADDJ:   IF ( MAXADD > 0 ) THEN GO TO BUMADD ;

        ROUTEX = PROMPTA ;
        NLINE = 1 ;
        MAXD = REFS (12) ;
        MAXR = REFS (11) ;

        ROUTE = STEP30 ;
             GO TO FINDASLASH ;

STEP30: IF ( J1 * J2 * J3 * J4 ) = 0 THEN GO TO BUMENTRY ;
```

```
PAGE NUMBER :  10

SOURCE CARD LISTING FOR
 P5B:    /* INTERACTIVE SEARCH RESULTS EDITOR - WED/09/MAR/77 */

        NTYP = 3 ;  NSTAT = 2 ;  NDSTAT = 2 ;
        NRATE = SUBSTR ( CARD, (J1 + 1), (J2 - J1 - 1) ) ;
        NRANK = SUBSTR ( CARD, (J2 + 1), (J3 - J2 - 1) ) ;
        NJTASK = SUBSTR ( CARD, (J3 + 1), (J4 - J3 - 1) ) ;
        YETASK = NID ;

        NADD = NADD + 1 ;
        IF ( NADD > 500 ) THEN GO TO BUMADD ;
        NREC = TOTREC + NADD ;

        DLINE = 'THE FOLLOWING RECORD HAS BEEN ADDED...' ;
        DISPLAY ( '     ' ) ;  DISPLAY ( DLINE ) ;

        PUT STRING (DLINE) EDIT
             ( (10)' ', 'RECNO/RATE/RANK/JOB : ', NREC,
              '/', NRATE, '/', NRANK, '/', NJTASK )
             ( A, A, P'ZZ,ZZ9', 6 ( A )  ) ;

        LOOKU = STEP31 ;
             GJ TO FINDTASK ;

STEP31: IF ( TASKREC > 0 ) THEN GO TO STEP32 ;
             SJBSTR ( DLINE, 56 ) = '(NO MATCHING DATA)' ;
                  GO TO ADDOUT ;

STEP32: IF ( TASKREC > MAXR ) THEN GO TO ER5 ;

        READ FILE (TASKS) INTO (ATASKS) KEY (TASKREC) ;

        CARD = ' ' ;
        SUBSTR ( CARD, 20 ) = TASKTTL ;
        NDSTAT = 1 ;
        NLINE = 2 ;

             GJ TO ADDOUT ;

/****************************************/

   /* THIS SUBROUTINE SEARCHES OUT THE IMBEDED SLASHES IN AN
      ADDITION-TYPE KEYBOARD ENTRY.
             POSSIBLE FORMATS : (1) ADDC / CIN / COP /
                                (2) ADDV /  VEH NUM  /
                                (3) ADDJ / RATE / RANK / JOBTASK /
             ON EXIT : J1, J2, J3, J4 SET AS REQD TO POINT TO
                           THE RELATED '/'
             ERROR RETURN TO BUMENTRY
             RETURN VECTOR IS ROUTE              */

FINDASLASH:
        J1, J2, J3, J4 = 0 ;
        DLINE = CARD ;

        J1 = INDEX (CARD, '/') ;
        IF ( J1 = 0 ) | ( J1 > 20 ) THEN GO TO BUMENTRY ;
        CTAB (J1) = ' ' ;

        J2 = INDEX ( CARD, '/' ) ;
        IF ( J2 <= J1 ) | ( J2 = 0 ) THEN GO TO BUMENTRY ;
        CTAB (J2) = ' ' ;

        J3 = INDEX ( CARD, '/' ) ;
        IF ( J3 = 0 ) THEN GO TO SLRET ;
        IF ( J3 <= J2 ) THEN GO TO BUMENTRY ;
        CTAB (J3) = ' ' ;

        J4 = INDEX ( CARD, '/' ) ;
        IF ( J4 = 0 ) THEN GO TO SLRET ;
        IF ( J4 <= J3 ) THEN GO TO BUMENTRY ;
```

```

SLRET:  CARD = DLINE ;

             GO TO ROUTE ;

/****************************************/

   /* HERE, IS SUBROUTINE TO DETERMINE LEGALITY OF AN INPUT
      RECORD NUMBER.
             ON ENTRY - NUMBER TO BE VERIFIED IS IN K
             ON EXIT  - NREC = SS TO RESULTS STRUCTURE
                        NRECA = SS TO ADDS STRUCTURE
             ERROR EXIT TO BUMENTRY
             RETURN VECTOR IS ROUTE                *.

FINDREC:
       IF ( K <= (TOTREC + NADD) ) THEN GO TO FINDR1 ;

BUMNUMBER:   DLINE = (5)' ' || '** RANGE ERROR IN RECORD NUMBER' ;
        DISPLAY ( '     ' ) ;  DISPLAY ( DLINE ) ;
                  GO TO PROMPTA ;

FINDR1: IF ( K <= TOTREC )
             THEN DO ;  NREC = K ;  NRECA = 0 ;  END ;
             ELSE DO ;  NREC = 0 ;  NRECA = K - TOTREC ;  END ;

                  GO TO ROUTE ;

/****************************************/

   /* THIS IS THE SUBROUTINE WHICH ISOLATES THE NUMERIC
      SINGLE/PAIRS FROM THE DELETE & PRINT KEYBOARD ENTRIES
             ON ENTRY - NO REQUIREMENTS
             ON EXIT  - RETURN VECTOR IS ROUTE
                      - FIRST NUMBER BIN IN N1, CHAR IN N1C
                      - 2ND NUM (IF ANY) BIN IN N2, CHAR IN N2C
                        2ND, IF ABSENT, N2 = 0
                      - NOS = 1 TO 2, INDICATES SING/PAIR INPUTS
             ERROR RETURN TO PROMPTA (BAD NUM) OR BUMENTRY      */

FINDNOS:
        N1, N2 = 0 ;
        N1C, N2C = ' ' ;
        NCONV = 2 ;
        NOS = 1 ;

        L1 = INDEX ( CARD, ' ' ) ;    /* FIND FIRST BLANK */

        IF ( L1 = 0 ) | ( L1 > 20 )
             | ( SUBSTR(CARD, L1) = ' ' )
             THEN GO TO BUMENTRY ;

LFN1:        DO L = L1 TO 80 ;         /* FIND FIRST DIGIT */
             L2 = L ;
             IF ( CTAB(L) ¬= ' ' ) THEN GO TO FINDN2 ;
             END LFN1 ;

                  GO TO BUMENTRY ;

FINDN2:      DO L = L2 TO 80 ;
             L1 = L ;
             IF ( CTAB(L) = ' ' ) | ( CTAB(L) = ',' )
                  THEN GO TO FINDN3 ;
             END FINDN2 ;

                  GO TO BUMENTRY ;

FINDN3: N1C = SUBSTR ( CARD, L2, (L1-L2) ) ;
        GET STRING (N1C) EDIT (N1) ( F(5) ) ;
```

```
PAGE NUMBER :  12
SOURCE CARD LISTING FOR
 P5B:     /* INTERACTIVE SEARCH RESULTS EDITOR - WED/09/MAR/77 */

           IF ( L1 > 79 ) THEN GO TO BUMENTRY ;

LFN2:            DO L = L1 TO 79 ;
                 L2 = L ;
                 IF ( CTAB(L) ¬= ' ' ) & ( CTAB(L) ¬= ',' )
                      THEN GO TO FINDN4 ;
                 END LFN2 ;

                      GO TO ROUTE ;

FINDN4: IF ( L2 > 75 ) THEN L3 = 80 ;
                       ELSE L3 = L2 + 5 ;

LFN3:            DO L = L2 TO L3 ;
                 L1 = L ;
                 IF ( CTAB(L) = ' ' ) THEN GO TO FINDN5 ;
                 END LFN3 ;

FINDN5: N2C = SUBSTR ( CARD, L2, (L1-L2) ) ;
        GET STRING (N2C) EDIT (N2) ( F(5) ) ;

           IF ( N2 < N1 ) THEN GO TO BUMENTRY ;

LFNOUT:          NOS = 2 ;
                      GO TO ROUTE ;

/**************************************/

   /* DIRECTORY LOOKUP SUBROUTINES IN THIS SECTION */

DECLARE ( FLCIN, FLCDP, FLVEH, FLTASK )    BIT (1) STATIC,
          YECIN             CHAR (8) STATIC,
          YECDP             CHAR (4) STATIC,
          ( YETASK, YEVEH )  CHAR (13) STATIC,
          ( CINREC, CDPREC, VEHREC, TASKREC, VEHMAX, TASKMAX,
               CINMAX, CDPMAX )    FIXED BIN STATIC,
          LOOKU             LABEL ;

   /* CIN DIRECTORY LOOKUP SUBROUTINE
               ENTRY : DESIRED CIN IN YECIN
               EXIT  : CINREC = 0 (NO FIND) OR REG NUMBER
               RETURN VECTOR IS LOOKU                 */

FINDCIN: IF ( FLCIN ) THEN GO TO CIN2 ;

         CINMAX = REFS (2) ;
         FLCIN = '1'B ;

CINLOAD:        DO J = 1 TO 4 ;
                PLCIN = ADDR ( COVCIN (J) ) ;
                READ FILE (DCIN) INTO (LOADCIN) ;
                END CINLOAD ;

CIN2:           DO I = 1 TO CINMAX ;
                IF ( YECIN > CHCIN(I) ) THEN GO TO ECIN2 ;
                IF ( YECIN ¬= CHCIN(I) ) THEN GO TO NFCIN ;
                     CINREC = CINPTR (I) ;
                          GO TO LOOKU ;
ECIN2:          END CIN2 ;

NFCIN:  CINREC = 0 ;
        GO TO LOOKU ;

   /* CDP DIRECTORY LOOKUP SUBROUTINE
               ENTRY : DESIRED CDP IN YECDP
               EXIT  : CDPREC = 0 (NO FIND), OR REC NUMBER
               RETURN VECTOR IS LOOKU                  */
```

```
PAGE NUMBER :  13

SOURCE CARD LISTING FOR
 P5B:    /* INTERACTIVE SEARCH RESULTS EDITOR - WED/09/MAR/77 */


FINDCDP: IF ( FLCDP ) THEN GO TO CDP2 ;

        CDPMAX = REFS (3) ;
        FLCDP = '1'B ;

CDPLOAD:     DO J = 1 TO 4 ;
             PLCDP = ADDR ( COVCDP (J) ) ;
             READ FILE (DCDP) INTO (LOADCDP) ;
             END CDPLOAD ;

CDP2:        DO I = 1 TO CDPMAX ;
             IF ( YECDP > CHCDP(I) ) THEN GO TO ECDP2 ;
             IF ( YECDP ¬= CHCDP(I) ) THEN GO TO NFCDP ;
                  CDPREC = CDPPTR(I) ;
                       GO TO LOOKU ;
ECDP2:       END CDP2 ;

NFCDP:  CDPREC = 0 ;
             GO TO LOOKU ;

   /* VEHICLE ID LOOKUP SUBROUTINE
             ENTRY : DESIRED VEH ID IN YEVEH
             EXIT  : VEHREC = 0 (NO FIND), OR REC NUMBER
             RETURN VECTOR IS LOOKU          */

FINDVEH: IF ( FLVEH ) THEN GO TO VEH2 ;

        VEHMAX = REFS (8) ;
        FLVEH = '1'B ;

VEHLOAD:     DO J = 1 TO 4 ;
             PLVEH = ADDR ( COVVEH (J) ) ;
             READ FILE (DVEH) INTO (LOADVEH) ;
             END VEHLOAD ;

VEH2:        DO I = 1 TO VEHMAX ;
             IF ( YEVEH > CHVEH(I) ) THEN GO TO EVEH2 ;
             IF ( YEVEH ¬= CHVEH(I) ) THEN GO TO NFVEH ;
                  VEHREC = VEHPTR (I) ;
                       GO TO LOOKU ;
EVEH2:       END VEH2 ;

NFVEH:  VEHREC = 0 ;
             GO TO LOOKU ;

   /* TASK ID DIRECTORY LOOKUP SUBROUTINE
             ENTRY : DESIRED TASK ID (13-CHAR) IN YETASK
             EXIT  : TASKREC = 0 (NO FIND) OR REC NUMBER
             RETURN VECTOR IS LOOKU          */

FINDTASK: IF ( FLTASK ) THEN GO TO TASK2 ;

          TASKMAX = REFS (12) ;
        FLTASK = '1'B ;

TASKLOAD:    DO J = 1 TO 12 ;
             PLTASK = ADDR ( COVTASK (J) ) ;
             READ FILE (DTASK) INTO (LOADTASK) ;
             END TASKLOAD ;

TASK2:       DO I = 1 TO TASKMAX ;
             IF ( YETASK > CHTASK(I) ) THEN GO TO ETASK2 ;
             IF ( YETASK ¬= CHTASK(I) ) THEN GO TO NFTASK ;
                  TASKREC = TASKPTR (I) ;
                       GO TO LOOKU ;
ETASK2:      END TASK2 ;

NFTASK: TASKREC = 0 ;
```

```
             GO TO LOOKU ;

/****************************************/

   /* TERMINAL, IE, CATASTROPHIC BADNESSES REPORTED HERE */

BUSTED: PUT STRING (CARD) EDIT ( (10)' ', '** ERROR NUMBER ',
             NER, ' HAS OCCURED **' )
             ( A, A, F(2), A ) ;

        DISPLAY ( '    ' ) ;  DISPLAY ( CARD ) ;
        DISPLAY ( '    ' ) ;  DISPLAY ( DLINE ) ;

/****************************************/

   /* HERE, REE RESULTS PROJECT FILE IS COMPOSED FROM THE INPUT
      RESULTS ARRAY AND ANY ADDS ARRAY ENTRIES */

QUIT:   OPEN FILE (RESU) RECORD OUTPUT,
             FILE (REE) RECORD OUTPUT ;

RELOADR:     DO J = 1 TO TOTREC ;
             S16 = RESREC (J) ;
             WRITE FILE (RESU) FROM (S16) ;

             IF ( ROSTAT(J) = 3 ) THEN GO TO ELOADR ;
             WRITE FILE (REE) FROM (S16) ;
                  NEE = NEE + 1 ;

ELOADR:      END RELOADR ;

        IF ( NADD <= 0 ) THEN GO TO EOJ ;

RELOADA:     DO J = 1 TO NADD ;

             IF ( MSTAT(J) = '3' ) | ( MTYP(J) = '0' )
                  THEN GO TO ERLOADA ;

             NEW = ADDS (J) ;
             WRITE FILE (RESU) FROM (NEW) ;

             WRITE FILE (REE) FROM (NEW) ;
                  NEE = NEE + 1 ;

ERLOADA:     END RELOADA ;

                  GO TO EOJ ;

/****************************************/

   /* HERE WE ARE AT THE LAST ROUNDUP */

EOJ:    DLINE = '** ROE SEARCH RESULTS EDITOR IS TERMINATING' ;
        DISPLAY ( '    ' ) ;  DISPLAY ( DLINE ) ;

        PUT STRING (DLINE) EDIT ( '    TOTAL RECORDS ADDED : ', NADD )
             ( A, F(3) ) ;
        DISPLAY ( '    ' ) ;  DISPLAY ( DLINE ) ;

        PUT STRING (DLINE) EDIT (
             '    TOTAL RECORDS IN PROJECT EXTRACT FILE : ',
             NEE ) ( A, P'Z,ZZ9' ) ;
        DISPLAY ( '    ' ) ;  DISPLAY ( DLINE ) ;

        CLOSE FILE (MREF), FILE (ID),
              FILE (COURSE), FILE (DCIN), FILE (DCOP),
              FILE (VEHS), FILE (DVEH),
              FILE (TASKS), FILE (DTASK),
              FILE (RESU), FILE (REE) ;
```

```
PAGE NUMBER :  15

SOURCE CARD LISTING FOR
 P5B:    /* INTERACTIVE SEARCH RESULTS EDITOR - WED/09/MAR/77 */

966
967                  END P5B ;
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
 P5C:    /* PRINT OF ROE SEARCH RESULTS FOR EXTRACT - FRI/11/MAR/77 */


P5C:    /* PRINT OF ROE SEARCH RESULTS FOR EXTRACT - FRI/11/MAR/77 */
        PROC (INPARM) OPTIONS (MAIN) ;

   /*   REVISIONS :
        FRI/11/MAR/77 - VERYFIRST IMPLEMENTATION.
                      - NOTE: THIS IS A PERMUTATION OF P5A.

        THU/24/MAR/77 - MODS PRIOR TO TRANSMIT TO NCSS.
   */

DECLARE DLINE8         CHAR (8).  DEF DLINE POS (8),
        DLINECST       CHAR (36) DEF DLINE POS (34),
        DLINE28        CHAR (4)  DEF DLINE POS (28),
        DLINE33        CHAR (36) DEF DLINE POS (33),
        DLINE36        CHAR (36) DEF DLINE POS (36),
        DLINE40        CHAR (40) DEF DLINE POS (40),
         CARD7         CHAR (70) DEF CARD POS (8),

        NODIR          CHAR (35) STATIC INIT
             ( '** DIRECTORY SEARCH WAS NEGATIVE **' ),

        NOAB           CHAR (36) STATIC INIT
             ( '(NO ABBREVIATED FILE DATA AVAILABLE)' ) ;
DECLARE 1 TTLCARD      STATIC,        /* PROJECT TITLE DATA BASE */
          2 PNAME      CHAR (10),     /* CKF.ID */
          2 PTTL       CHAR (70),

        INPARM         CHAR (100) VARYING,
        1 POVER        BASED (PP),
          2 PARM2      CHAR (4),

        ( DISP, PTR, DPAGE, I, J, K, L, LT, NC, NFIRST,
             NN, NPAGE, NR, NTYPE, NV, NER, NLINE,
             NDF, NJ ) FIXED BIN STATIC,

        DTTL           CHAR (23) STATIC INIT
             ( '       PAGE NUMBER : XXXX' ),
        DDPGE          PICTURE 'ZZZ9' DEF DTTL POS (20),

        MAXLTB         FIXED BIN STATIC INIT ( 60 ) ;

DECLARE ( COURSE, VEHS, TASKS
             ) FILE RECORD DIRECT KEYED ENV ( REGIONAL(1) ) ;

DECLARE ( CARD, DLINE )     CHAR (80) STATIC,
        C110           CHAR (10) DEF CARD,
        C1180          CHAR (70) DEF CARD POS (11) ;

DECLARE 1 MAST         STATIC,
          2 REFS (40)  FIXED BIN,

        1 RESULT       BASED (PR),
          2 RTYP       PICTURE '9',
          2 RSTAT      PICTURE '9',
          2 RDSTAT     PICTURE '9',
          2 RESID      CHAR (13),

        1 RESC         BASED (PR),
          ( 2 F1, 2 F2, 2 F3 )    CHAR (1),
          2 RESCIN     CHAR (8),
          2 RESCDP     CHAR (4),

        1 REST         BASED (PR),
          ( 2 F1, 2 F2, 2 F3 )    CHAR (1),
          2 RATE       CHAR (5),
          2 RANK       CHAR (2),
          2 JOBTASK    CHAR (6),

PAGE NUMBER :    2

SOURCE CARD LISTING FOR
 P5C:    /* PRINT OF ROE SEARCH RESULTS FOR EXTRACT - FRI/11/MAR/77 */

        S16            CHAR (16) STATIC,
        V1 (3)         LABEL ;

DECLARE T1             CHAR (32) STATIC INIT (
'RECNO    CIN    CDP        COURSE' ),

        T1U            CHAR (37) STATIC INIT (
'------ -------- ---- ----------------' ),

        T2             CHAR (46) STATIC INIT (
'RECNO  STOCK NUMBER  FILE NUM  TRAINING DEVICE' ),

        T2U            CHAR (46) STATIC INIT (
'------ ------------- --------- ---------------' ),

        T3             CHAR (29) STATIC INIT (
'RECNO  RATE  RANK  JOB   TASK' ),

        T3U            CHAR (29) STATIC INIT (
'------ ----- ---- ------ ----' ) ;

DECLARE WINDUPS (6)    CHAR (44) STATIC INIT (
             '     TOTAL NUMBER COURSE RECORDS PROCESSED : ',
             '    TOTAL NUMBER VEHICLE RECORDS PROCESSED : ',
             '    TOTAL NUMBER JOBTASK RECORDS PROCESSED : ',
             '       TOTAL NUMBER ROE RESULT RECORDS READ : ',
             'RECORDS WITHOUT MATCHING ABBREVIATED DATA : ',
             '    TOTAL NUMBER DIRECTORY SEARCH FAILURES : ' ),

        WIND (6)       FIXED BIN STATIC ;


/************************************/

DECLARE 1 ACRS         STATIC,        /* ABBR. COURSE FILE */
          2 SET1,                     /* CKF.ACOURSES */
            3 CDP      CHAR (4),
            3 CIN      CHAR (8),
            3 CST      CHAR (16),
            3 NOBC     CHAR (4),
            3 VEC      CHAR (4),
          2 PC         CHAR (2),
          2 SET2,
            3 RMS      CHAR (3),
            3 TYCRS    CHAR (2),
            3 SVC      CHAR (1),
            3 MI       CHAR (1),
          2 STCD       CHAR (1),
          2 STDTE      PICTURE '(5)9',
          2 SET3,
            3 TRAPS    CHAR (1),
            3 TPC      CHAR (5),
          2 ATTR       PICTURE '99V9',
          2 STBK       PICTURE '99V9',
          2 CLEN       PICTURE '999',
          2 THRS       PICTURE '999',
          2 LHRS       PICTURE '999',
          2 CAJB       PICTURE '(5)9V99',
          2 CC80       CHAR (1) ;

DECLARE 1 CINDER (304) STATIC,        /* CRS CIN DIRECTORY */
          2 CHCIN      CHAR (8),      /* CKF.CRS.DIRCIN */
          2 CINPTR     FIXED BIN,

        LOADCIN        CHAR (760)  BASED (PLCIN),
        COVCIN (4)     CHAR (760)  BASED (PTCIN) ;

DECLARE 1 CDPDIR (1200) STATIC,       /* CRS CDP DIRECTORY */
          2 CHCDP      CHAR (4),      /* CKF.CRS.DIRCDP */
```

```
AGE NUMBER :   3

SOURCE CARD LISTING FOR
 PSC:     /* PRINT OF ROE SEARCH RESULTS FOR EXTRACT - FRI/11/MA

           2 CDPPTR        FIXED BIN,

        LOADCDP            CHAR (1200) BASED (PLCDP),
        COVCDP (6)         CHAR (1200) BASED (PTCDP) ;

/****************************************/

DECLARE 1 AVEHICLES        STATIC,           /* ABBR. VEH FILE */
           2 STOCKN        CHAR (13),        /* CKF.AVEHS */
           2 DEVDESIG      CHAR (9),
           2 DEVNAME       CHAR (47),
           2 DEVCOST       PICTURE '(8)9V99',
           2 CC80          CHAR (1),

        VCOST              FIXED DEC (10,2) STATIC,
        VOVER              CHAR (80) BASED (PV80) ;

DECLARE 1 VEHDIR (200)     STATIC,           /* VEHICLE DIRECTORY */
           2 CHVEH         CHAR (13),        /* CKF.VEH.DIR */
           2 VEHPTR        FIXED BIN,

        COVVEH (4)         CHAR (750) BASED (PTVEH),
        LOADVEH            CHAR (750) BASED (PLVEH) ;

/****************************************/

DECLARE 1 ATASKS           STATIC,           /* ABBR. TASK FILE */
           2 RATING        CHAR (7),         /* CKF.ATASKS */
           2 JOBTASK       CHAR (6),
           2 TASKTTL       CHAR (50),
           2 FILL          CHAR (6),
           2 BILCOST       PICTURE '(8)9V99',
           2 CC80          CHAR (1) ;

DECLARE 1 TASKDIR (1200) STATIC,             /* JOBTASK FILE DIRECTORY */
           2 CHTASK        CHAR (13),        /* CKF.TASKS.DIR */
           2 TASKPTR       FIXED BIN,

        COVTASK (12)       CHAR (1500) BASED (PTTASK),
        LOADTASK           CHAR (1500) BASED (PLTASK) ;

/****************************************/

START:  NTYPE, NC, NJ, NV, NR, NN, NDF,
             NPAGE, DPAGE, LT, NFIRST,
             J, K = 0 ;

        FLCIN, FLCDP, FLVEH, FLTASK = '0'B ;

        PP = ADDR (INPARM) ;
        PR = ADDR (S16) ;
        PTCIN = ADDR (CINDER) ;
        PTCDP = ADDR (CDPDIR) ;
        PV80 = ADDR (AVEHICLES) ;
        PTVEH = ADDR (VEHDIR) ;
        PTTASK = ADDR (TASKDIR) ;

COMMENCE:
        OPEN FILE (MREF) RECORD INPUT,
             FILE (ID) RECORD INPUT,
             FILE (COURSE) INPUT,
             FILE (DCIN) RECORD INPUT,
             FILE (DCDP) RECORD INPUT,
             FILE (VEHS) INPUT,
             FILE (DVEH) RECORD INPUT,
             FILE (TASKS) INPUT,
             FILE (DTASK) RECORD INPUT,
             FILE (REE) RECORD INPUT ;
```

```
PAGE NUMBER :    4

 SOURCE CARD LISTING FOR
  P5C:     /* PRINT OF ROE SEARCH RESULTS FOR EXTRACT - FRI/11/MAR/77

        ON ENDFILE (REE) GO TO EOJ ;

        READ FILE (MREF) INTO (MAST) ;

/***************************************/

   /* HERE, DEVICE ASSIGNMENTS ARE VERIFIED */

FIRST:  DISP, PTR = 1 ;
        IF ( PARM2 = 'BOTH' ) THEN GO TO STEPA ;

        IF ( PARM2 = 'TERM' )
             THEN PTR = 2 ;
             ELSE DISP = 2 ;

        IF ( PARM2 = 'NONE' )
             THEN PTR, DISP = 1 ;

   /* HERE, PRINTER DATASET IS INITIALIZED */

STEPA:  IF ( PTR = 2 ) THEN GO TO STEPB ;
        OPEN FILE (SYSPRINT)
             LINESIZE (120) PAGESIZE (60) ;

        ON ENDPAGE (SYSPRINT)
             BEGIN ;
             NPAGE = NPAGE + 1 ;

             IF ( NFIRST = 1 ) THEN
                  DO ;  PUT PAGE LINE (5) ;  GO TO TSTTTL ;  END ;
             PUT SKIP (3) ;
             NFIRST = 1 ;

TSTTTL:      IF ( NTYPE = 3 ) THEN GO TO TASKTTL ;
             IF ( NTYPE = 2 ) THEN GO TO VEHTTL ;

CRSTTL:      PUT EDIT (
'*** EXTRACT DATA IS REQUESTED FOR THE FOLLOWING COURSES ***',
                  'PAGE NUMBER : ', NPAGE, 'RECORD',
                  ' NUM    ORIGIN      CIN      CDP    COURSE TITLE',
                  '-----  --------  --------   ----  -----------------'.
                  '     ' )
                                 ( X(9), A, SKIP(2), X(9), A, F(3),
                    SKIP(2), X(4), A, 3 ( SKIP(1), A ) ) ;

                       GO TO SKP ;

VEHTTL:      PUT EDIT (
'*** EXTRACT DATA IS REQUESTED FOR THE FOLLOWING VEHICLES ***',
                  'PAGE NUMBER : ', NPAGE, 'RECORD', 'DEVICE',
                  ' NUM    ORIGIN     STOCK NUMBER  DESIGNATOR',
                  '-----  --------   ------------  ----------',
                  '     ' )
                                 ( X(9), A, SKIP(2), X(9), A, F(3),
                    SKIP(2), X(4), A, COL(35), A, 3 ( SKIP(1), A ) ) ;

                       GO TO SKP ;

TASKTTL:     PUT EDIT (
'*** EXTRACT DATA IS REQUESTED FOR THE FOLLOWING JOBS / TASKS ***',
                  'PAGE NUMBER : ', NPAGE, 'RECORD',
                  ' NUM    ORIGIN   RATING  RANK  JOBTASK',
                  '-----  --------  ------  ----  -------', '     ' )
                                  ( X(9), A, SKIP(2), X(9), A, F(3),
                   SKIP(2), X(4), A, 3 ( SKIP(1). A ) ) ;

SKP:         PUT SKIP (1) ;
             END ;
```

```
PAGE NUMBER :    5

SOURCE CARD LISTING FOR
 15C:    /* PRINT OF ROE SEARCH RESULTS FOR EXTRACT - FRI/11/MAR/77

   /* HERE, INITILIZATION OF TERMINAL DATASET PROCEEDS */

STEP3:  IF ( DISP = 2 ) THEN GO TO STEPC ;

        ON CONDITION (NEWDP)
            BEGIN ;
            IF ( LT < 61 ) THEN LT = 61 ;

            IF ( LT >= 66 ) THEN GO TO LPB ;

            IF ( LT < 61 ) THEN LT = 61 ;
LPA:            DO L = LT TO 66 ;
                DISPLAY ( '     ' ) ;
                END LPA ;

LPB:        DISPLAY ( '     ' ) ;

            DPAGE = DPAGE + 1 ;
            DDPGE = DPAGE ;
            DISPLAY ( DTTL ) ;
            DISPLAY ( '     ' ) ;
            LT = 3 ;
            END ;

   /* NOW, MASTER TITLE STRIP IS OUTPUT */

STEPC:  READ FILE (ID) INTO (CARD) ;

        DLINE = (10)' ' || '** EXTRACT SEARCH RESULTS FOR PROJECT - '
            || C110 || ' **' ;
        CARD = (12)' ' || C1180 ;

        DISPLAY ( '     ' ) ;   DISPLAY ( DLINE ) ;
        DISPLAY ( CARD ) ;   DISPLAY ( '     ' ) ;

        /* DISPLAY OUTPUT GOES HERE */

        IF ( PTR = 2 ) THEN GO TO RD1 ;

        PUT EDIT ( DLINE, CARD ) ( SKIP(3), A, SKIP(1), A ) ;

            GO TO RD1 ;

/****************************************/

   /* HEREIN, IS START OF MAINLINE LOOP */

RD1:    READ FILE (REE) INTO (S16) ;
        NR = NR + 1 ;

        IF ( RTYP = NTYPE ) THEN GO TO STEP1 ;

            NTYPE = RTYP ;
            IF ( PTR = 1 ) THEN SIGNAL ENDPAGE (SYSPRINT) ;
            IF ( DISP = 1 ) THEN SIGNAL CONDITION (NEWDP) ;

STEP1:  GO TO V1 (NTYPE) ;

   /* HERE, A COURSE RECORD IS PROCESSED */

V1(1):  NC = NC + 1 ;
        NLINE = 1 ;

        DLINE = ' ' ;
        PUT STRING (DLINE) EDIT ( NR, RESCIN, RESCDP )
            ( P'Z,ZZ9', X(12), A, X(2), A ) ;

        IF ( RSTAT = 1 ) | ( RSTAT = 2 ) THEN GO TO STP1 ;
ER1:        NER = 1 ;
```

```
PAGE NUMBER :   6

SOURCE CARD LISTING FOR
 P5C:     /* PRINT OF ROE SEARCH RESULTS FOR EXTRACT - FRI/11/MAR/77 */

                  PUT STRING (DLINE) DATA ( NR, S16 ) ;
                        GO TO BUSTED ;

STP1:    IF ( RSTAT = 1 ) THEN DLINE8 = 'ORIGINAL' ;
                             ELSE DLINE8 = ' ADDED' ;

         IF ( RESCDP = ' ' ) THEN DLINE28 = 'NONE' ;

         IF ( RDSTAT = 2 ) | ( RESCDP = ' ' ) THEN
NOCDATA:       DO ;
               NN = NN + 1 ;
               DLINECST = NOAB ;
                     GO TO OUTLOOP ;
               END NOCDATA ;

         YECDP = RESCDP ;
         LOOKU = STEP2 ;
               GO TO FINDCDP ;

STEP2:   IF ( CDPREC = 0 ) THEN
LP1:           DO ;
               NDF = NDF + 1 ;
               DLINECST = NODIR ;
                           GO TO OUTLOOP ;
               END LP1 ;

         READ FILE (COURSE) INTO (ACRS) KEY (CDPREC) ;

         DLINECST = CST ;

                     GO TO OUTLOOP ;

      /* HERE, VEHICLE TYPE RECORDS ARE PROCESSED */

V1(2):   NV = NV + 1 ;
         NLINE = 1 ;

         DLINE = ' ' ;
         PUT STRING (DLINE) EDIT ( NR, RESID )
               ( P'Z,ZZ9', X(12), A(13) ) ;

         IF ( RSTAT = 1 ) | ( RSTAT = 2 ) THEN GO TO STP2 ;
                                        ELSE GO TO ER1 ;

STP2:    IF ( RSTAT = 1 ) THEN DLINE8 = 'ORIGINAL' ;
                          ELSE DLINE8 = ' ADDED' ;

         IF ( RDSTAT = 2 ) THEN
NOVDATA:       DO ;
               NN = NN + 1 ;
               DLINE33 = NOAB ;
                           GO TO OUTLOOP ;
               END NOVDATA ;

         YEVEH = RESID ;
         LOOKU = STEP3 ;
               GO TO FINDVEH ;

STEP3:   IF ( VEHREC = 0 ) THEN
LP2:           DO ;
               NDF = NDF + 1 ;
               DLINE36 = NODIR ;
                           GO TO OUTLOOP ;
               END LP2 ;

         READ FILE (VEHS) INTO (AVEHICLES) KEY (VEHREC) ;

         DLINE33 = DEVDESIG ;
         CARD = ' ' ;
```

SOURCE CARD LISTING FOR
P5C: /* PRINT OF ROE SEARCH RESULTS FOR EXTRACT - FRI/11/MAR/77 */

```
          CARD7 = DEVNAME ;
          NLINE = 2 ;

                        GO TO OUTLOOP ;

   /* NOW, PROCESS A TASK RECORD HERE */

V1(3):    NJ = NJ + 1 ;
          NLINE = 1 ;

          DLINE = ' ' ;
          PUT STRING (DLINE) EDIT ( NR, RATE, RANK, REST.JOBTASK )
                ( P'Z,ZZ9', X(12), A, X(4), A, X(3), A ) ;

          IF ( RSTAT = 1 ) THEN GO TO STP3 ;
          IF ( RSTAT = 2 ) THEN GO TO STP4 ;
                GO TO ER1 ;

STP3:     DLINE8 = 'ORIGINAL' ;
                GO TO STP5 ;
STP4:     DLINE8 = ' ADDED' ;

STP5:     IF ( RDSTAT = 2 ) THEN
NOTDATA:        DO ;
                NN = NN + 1 ;
                DLINE40 = NOAB ;
                      GO TO OUTLOOP ;
                END NOTDATA ;

          YETASK = RESID ;
          LOOKU = STEP4 ;
                GO TO FINDTASK ;

STEP4:    IF ( TASKREC = 0 ) THEN
LP3:            DO ;
                NDF = NDF + 1 ;
                DLINE40 = NODIR ;
                      GO TO OUTLOOP ;
                END LP3 ;

          READ FILE (TASKS) INTO (ATASKS) KEY (TASKREC) ;

          CARD = ' ' ;
          CARD7 = TASKTTL ;
          NLINE = 2 ;

                GO TO OUTLOOP ;

   /* HERE IS THE PRINTED LINE OUTPUT SEQUENCE */

OUTLOOP: IF ( PTR = 1 )
                THEN PUT LIST (DLINE) SKIP (2) ;

          IF ( PTR = 1 ) & ( NLINE = 2 )
                THEN PUT LIST ( CARD ) SKIP (1) ;

          IF ( DISP = 2 ) THEN GO TO RD1 ;

          IF ( LT > MAXLTB ) THEN SIGNAL CONDITION (NEWDP) ;

          DISPLAY ( DLINE ) ;
          LT = LT + 1 ;

          IF ( NLINE ¬= 2 ) THEN GO TO RD1 ;
          DISPLAY ( CARD ) ;
          LT = LT + 1 ;

                GO TO RD1 ;
```

```
PAGE NUMBER :    8

SOURCE CARD LISTING FOR
 P5C:    /* PRINT OF ROE SEARCH RESULTS FOR EXTRACT - FRI/11/MAR/77 */

/**************************************/

DECLARE ( FLCIN, FLCDP, FLVEH, FLTASK )      BIT (1) STATIC,
        YECIN               CHAR (8) STATIC,
        YECDP               CHAR (4) STATIC,
        ( YETASK, YEVEH )   CHAR (13) STATIC,
        ( CINREC, CDPREC, VEHREC, TASKREC, VEHMAX, TASKMAX,
             CINMAX, CDPMAX )    FIXED BIN STATIC,
        LOOKU               LABEL ;

   /* CIN DIRECTORY LOOKUP SUBROUTINE

             ENTRY : DESIRED CIN IN YECIN
             EXIT  : CINREC = 0 (NO FIND) OR REG NUMBER
             RETURN VECTOR IS LOOKU           */

FINDCIN: IF ( FLCIN ) THEN GO TO CIN2 ;

        CINMAX = REFS (2) ;
        FLCIN = '1'B ;

CINLOAD:     DO J = 1 TO 4 ;
             PLCIN = ADDR ( COVCIN (J) ) ;
             READ FILE (DCIN) INTO (LOADCIN) ;
             END CINLOAD ;

CIN2:        DO I = 1 TO CINMAX ;
             IF ( YECIN ¬= CHCIN(I) ) THEN GO TO ECIN2 ;
                  CINREC = CINPTR (I) ;
                       GO TO LOOKU ;
ECIN2:       END CIN2 ;

NFCIN:  CINREC = 0 ;
             GO TO LOOKU ;

   /* CDP DIRECTORY LOOKUP SUBROUTINE
             ENTRY : DESIRED CDP IN YECDP
             EXIT  : CDPREC = 0 (NO FIND), OR REC NUMBER
             RETURN VECTOR IS LOOKU           */

FINDCDP: IF ( FLCDP ) THEN GO TO CDP2 ;

        CDPMAX = REFS (3) ;
        FLCDP = '1'B ;

CDPLOAD:     DO J = 1 TO 4 ;
             PLCDP = ADDR ( COVCDP (J) ) ;
             READ FILE (DCDP) INTO (LOADCDP) ;
             END CDPLOAD ;

CDP2:        DO I = 1 TO CDPMAX ;
             IF ( YECDP ¬= CHCDP(I) ) THEN GO TO ECDP2 ;
                  CDPREC = CDPPTR(I) ;
                       GO TO LOOKU ;
ECDP2:       END CDP2 ;

NFCDP:  CDPREC = 0 ;
             GO TO LOOKU ;

   /* VEHICLE ID LOOKUP SUBROUTINE
             ENTRY : DESIRED VEH ID IN YEVEH
             EXIT  : VEHREC = 0 (NO FIND), OR REC NUMBER
             RETURN VECTOR IS LOOKU           */

FINDVEH: IF ( FLVEH ) THEN GO TO VEH2 ;

        VEHMAX = REFS (8) ;
        FLVEH = '1'B ;
```

```
PAGE NUMBER :   9

SOURCE CARD LISTING FOR
 P5C:    /* PRINT OF ROE SEARCH RESULTS FOR EXTRACT - FRI/11/MAR/77 */

VEHLOAD:  DO J = 1 TO 4 ;
          PLVEH = ADDR ( COVVEH (J) ) ;
          READ FILE (DVEH) INTO (LOADVEH) ;
          END VEHLOAD ;

VEH2:     DO I = 1 TO VEHMAX ;
          IF ( YEVEH ¬= CHVEH(I) ) THEN GO TO EVEH2 ;
              VEHREC = VEHPTR (I) ;
                  GO TO LOOKU ;
EVEH2:    END VEH2 ;

NFVEH:  VEHREC = 0 ;
          GO TO LOOKU ;

  /* TASK ID DIRECTORY LOOKUP SUBROUTINE
          ENTRY : DESIRED TASK ID (13-CHAR) IN YETASK
          EXIT  : TASKREC = 0 (NO FIND) OR REC NUMBER
          RETURN VECTOR IS LOOKU         */

FINDTASK: IF ( FLTASK ) THEN GO TO TASK2 ;

        TASKMAX = REFS (12) ;
      FLTASK = '1'B ;

TASKLOAD: DO J = 1 TO 12 ;
          PLTASK = ADDR ( COVTASK (J) ) ;
          READ FILE (DTASK) INTO (LOADTASK) ;
          END TASKLOAD ;

TASK2:    DO I = 1 TO TASKMAX ;
          IF ( YETASK ¬= CHTASK(I) ) THEN GO TO ETASK2 ;
              TASKREC = TASKPTR (I) ;
                  GO TO LOOKU ;
ETASK2:   END TASK2 ;

NFTASK: TASKREC = D ;
          GO TO LOOKU ;

/***************************************/

  /* IF GET HERE, PROGRAM IS TERRIBLE BROKE */

BUSTED: PUT STRING (CARD) EDIT
          ( '     ** ERROR NUMBER ', NER, ' HAS OCCURED **' )
          ( A, F(2), A ) ;

      IF ( PTR = 1 ) THEN PUT EDIT ( CARD, DLINE )
          ( SKIP(3), A, SKIP(2), A ) ;

      DISPLAY ( '    ' ) ;  DISPLAY ( DLINE ) ;
      DISPLAY ( '    ' ) ;   DISPLAY ( CARD ) ;

/***************************************/

  /* HERE IS EOF WINDUP PROCESSING */

EOJ:
EOF:  WIND(1) = NC ;      WIND(2) = NV ;
      WIND(3) = NJ ;      WIND(4) = NR ;
      WIND(5) = NN ;      WIND(6) = NDF ;

      IF ( PTR = 1 ) THEN PUT SKIP (3) ;

      DISPLAY ( '    ' ) ;
      DISPLAY ( '** NOW AT EOF ON SEARCH RESULTS FILE **' ) ;
      DISPLAY ( '    ' ) ;

SUMMARY:  DO J = 1 TO 6 ;
```

```
    PUT STRING (DLINE) EDIT
        ( WINDUPS(J), WIND(J) ) ( A(44), P'ZZ,ZZ9' ) ;

    IF ( PTR = 1 )
        THEN PUT LIST ( (10)' ' || DLINE ) SKIP (1) ;

    DISPLAY ( DLINE ) ;

    END SUMMARY ;

CLOSE FILE (MREF), FILE (ID), FILE (REE),
      FILE (COURSE), FILE (DCDP),
      FILE (VEHS), FILE (DVEH),
      FILE (TASKS), FILE (DTASK) ;

IF ( PTR = 1 ) THEN CLOSE FILE (SYSPRINT) ;

DISPLAY ( '     ' ) ;
    END P5C ;
```

```
PAGE NUMBER :   1

SOURCE CARD LISTING FOR
 P7:      /* ABBREVIATED COURSE FILE DIRECTORY CONSTRUCT - 12/14/76 */

P7:      /* ABBREVIATED COURSE FILE DIRECTORY CONSTRUCT - 12/14/76 */
         PROC (INPARM) OPTIONS (MAIN) ;

   /*    REVISIONS :
         TUE/14/DEC/76 - VERYFIRST IMPLEMENTATION.

         MON/14/MAR/77 - MODS TO PREPARE FOR SHIP TO NCSS.
   */

DECLARE INPARM           CHAR (100) VARYING,
        PARM             CHAR (1) BASED (PP),
        DLINE            CHAR (80) STATIC,
        YECIN            CHAR (8)
             STATIC INIT ( 'ABCDEFGH' ) ;

DECLARE ( N15, N35, N15R, NS, TOTCIN, TOTCDP, T35, LOOP, NT,
          NC,
          NFP, NER, J )     FIXED BIN STATIC ;

DECLARE COURSE   FILE RECORD INPUT,
        DCIN     FILE RECORD OUTPUT,
        DCDP     FILE RECORD OUTPUT ;

DECLARE IHESRTD          ENTRY
             ( CHAR (44), CHAR (27),
               FIXED BIN (31,0), FIXED BIN (31,0),
               ENTRY, ENTRY ),
        IHESARC          ENTRY ( FIXED BIN (31,0) ),
        E15A             ENTRY RETURNS ( CHAR (11) ),
        E35A             ENTRY,
        RCODE            FIXED BIN (31,0) ;

DECLARE RSORT            CHAR (20) STATIC,

        1 SORTREC        BASED (PSR),
          2 STYPE        CHAR (1),
          2 DFLD1        CHAR (4),
          2 DFLD2        CHAR (4),
          2 RECPTR       FIXED BIN,

        1 OVERD          BASED (POD),
          2 F1           CHAR (8),
          2 F2           FIXED BIN,

        1 OVERC          BASED (POC),
          2 F1           CHAR (4),
          2 F2           FIXED BIN ;

DECLARE 1 ACRS           STATIC,           /* ABBR. COURSE FILE */
          2 SET1,                          /* CKF.ACOURSES */
            3 CDP        CHAR (4),
            3 CIN        CHAR (8),
            3 CST        CHAR (16),
            3 NOBC       CHAR (4),
            3 NEC        CHAR (4),
          2 PC           CHAR (2),
          2 SET2,
            3 RMS        CHAR (3),
            3 TYCRS      CHAR (2),
            3 SVC        CHAR (1),
            3 MI         CHAR (1),
          2 STCD         CHAR (1),
          2 STDTE        PICTURE '(5)9',
          2 SET3,
            3 TRAPS      CHAR (1),
            3 TPC        CHAR (5),
          2 ATTR         PICTURE '99V9',
          2 STBK         PICTURE '99V9',
 PAGE NUMBER :   2

 SOURCE CARD LISTING FOR
  P7:     /* ABBREVIATED COURSE FILE DIRECTORY CONSTRUCT - 12/14/76 */

          2 CLEN        PICTURE '999',
          2 THRS        PICTURE '999',
          2 LHRS        PICTURE '999',
          2 CADB        PICTURE '(5)9V99',
          2 CCBO        CHAR (1) ;

DECLARE 1 CINDER (304) STATIC,          /* CRS CIN DIRECTORY */
          2 CHCIN       CHAR (8),       /* CKF.CRS.DIRCIN */
          2 CINPTR      FIXED BIN,
        LOADCIN         CHAR (760)  BASED (PLCIN),
        COVCIN (4)      CHAR (760)  BASED (PTCIN) ;

DECLARE 1 CDPDIR (1200) STATIC,         /* CRS CDP DIRECTORY */
          2 CHCDP       CHAR (4),       /* CKF.CRS.DIRCDP */
          2 CDPPTR      FIXED BIN,
        LOADCDP         CHAR (1200) BASED (PLCDP),
        COVCDP (6)      CHAR (1200) BASED (PTCDP) ;

DECLARE 1 MAST     STATIC,
          2 REFS (40)  FIXED BIN,
        CARD       CHAR (80) STATIC,
        MREF       FILE RECORD UPDATE ;

/****************************************/

   /* MASTER INITILIZATION DONE HERE, ONE TIME */

START:  PP = ADDR (INPARM) ;
        PTCIN = ADDR (CINDER) ;
        PTCDP = ADDR (CDPDIR) ;

        PSR = ADDR (RSORT) ;
        POC = ADDR (DFLD2) ;
        POD = ADDR (DFLD1) ;

        CHCIN = ' ' ;  CHCDP = ' ' ;
        CINPTR = 0 ;   CDPPTR = 0 ;

        N15, N35, N15R, NS, TOTCIN, TOTCDP, NT,
             NER = 0 ;
        T35, LOOP = 1 ;
        NC = -1 ;

        OPEN FILE (MREF) RECORD UPDATE ;
        READ FILE (MREF) INTO (MAST) ;

        DISPLAY ( '    ' ) ;
        DISPLAY ( 'ABBREVIATED COURSE DATABASE CIN/CDP DIRECTORY' ) ;
        DISPLAY ( '     CONSTRUCT PROGRAM' ) ;
        DISPLAY ( '    ' ) ;

        CALL IHESRTD (       /* BY REC TYPE, BY CIN-CDP */
             ' SORT FIELDS=(1,1,CH,A,2,8,CH,A),SIZE=E2000 ',
             ' RECORD TYPE=F,LENGTH=(20)',
             50000, RCODE, E15A, E35A ) ;

/****************************************/

E15A:   /* SORT RECORD MAKEUP PROCEDURE */
        PROC RETURNS ( CHAR (11) ) ;

        N15 = N15 + 1 ;

   /* THIS IS EOF ROUTINE FOR COURSE INPUT FILE */

        ON ENDFILE (COURSE)
             BEGIN ;

        DISPLAY ( '** EOF ON COURSE INPUT FILE ; '
 PAGE NUMBER :   3

 SOURCE CARD LISTING FOR
  P7:    /* ABBREVIATED COURSE FILE DIRECTORY CONSTRUCT - 12/14/76 */
             || 'SORT PHASE START' ) ;
        DISPLAY ( '     ' ) ;

             REFS(3), REFS(1) = NC ;

                  CALL IHESARC (8) ;   /* TELL SORT NOMORE RECORDS */
                       GO TO ENDE15A ;

             END ;

   /* E15A INITILIZATION DONE HERE ONLY ONE TIME */

        IF ( N15 > 1 ) THEN GO TO STEP1 ;

STEP1:  IF ( N15R > 0 ) THEN GO TO STEP3 ;

   /* THIS IS E15A PROC MAINLINE, PROCESS A NEW RECORD */

STEP2:  READ FILE (COURSE) INTO (ACRS) ;
        NC = NC + 1 ;
        RECPTR = NC ;
        STYPE = '1' ;

             OVERD.F1 = CIN ;
             N15R = 1 ;
                  GO TO GIVE ;

   /* HIRE, SECONDARY ENTRY: GIVE CDP RECORD OUT */

STEP3:  N15R = 0 ;
        STYPE = '2' ;

             OVERD.F1 = ' ' ;
            OVERC.F1 = CDP ;

   /* HERE, ANOTHER RECORD GIVEN TO SORT */

GIVE:   NS = NS + 1 ;
        CALL IHESARC ( 12 ) ;    /* GOT STILL ANOTHER RECORD */
        RETURN ( RSORT ) ;

ENDE15A:     END E15A ;

/***************************************/

E35A:   /* HERE IS THE SORT OUTPUT PROCEDURE */
        PROC ( INREC ) ;

DECLARE INREC     CHAR (20) ;

FM1:    FORMAT    ( SKIP(1), X(10), 2 ( X(2), F(4) ),
                    X(2), A(8), 2 ( X(2), F(4) ) ) ;

        N35 = N35 + 1 ;
        RSORT = INREC ;

   /* HERE, E35A INITILIZATION DONE ONE TIME */

        IF ( N35 > 1 ) THEN GO TO ROUTER ;

        DISPLAY ( '** SORT PHASE COMPLETE : STARTING CIN '
             || 'DIRECTORY MAKEUP' ) ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( 'CIN NUM     CIN      TOTAL CDPS   REC PTR' ) ;
        DISPLAY ( '-------   --------   ----------   -------' ) ;
        DISPLAY ( '     ' ) ;

   /* ENTRY ROUTING DONE HERE */
```

```
PAGE NUMBER :    4

SOURCE CARD LISTING FOR
 P7:     /* ABBREVIATED COURSE FILE DIRECTORY CONSTRUCT - 12/14/76 */

ROUTER: IF ( STYPE = '2' ) THEN GO TO TYPE2 ;

   /* HERE, STILL PROCESSING CIN ENTRIES */

TYPE1:  IF ( OVERD.F1 = YECIN ) THEN
DL1:         DO ;
             NT = NT + 1 ;
                  GO TO ANOTHER ;
             END DL1 ;

        IF ( N35 > 1 ) THEN      /* HERE, CINS HAVE CHANGED */
DL2:         DO ;
        PUT STRING (DLINE) EDIT
             ( TOTCIN, YECIN, NT, NFP )
             ( X(1), F(4), X(4), A(8), X(6), F(3), X(7), F(3) ) ;
        DISPLAY ( DLINE ) ;
             END DL2 ;

        NT = 1 ;
        YECIN = OVERD.F1 ;
        NFP = RECPTR ;
        TOTCIN = TOTCIN + 1 ;
        LOOP = LOOP + 1 ;

        IF ( LOOP > 304 ) THEN GO TO BUMCIN ;

             CINDEP (LOOP) = OVERD ;

ANOTHER:     CALL IHESARC ( 4 ) ;  /* GIMMIE ANOTHER SORT RECORD */
             RETURN ;

   /* HERE, STARTING CDP ENTRIES */

TYPE2:  IF ( T35 = 1 ) THEN      /* MUST FINISH LAST CIN ENTRY */
LP3:         DO ;

        PUT STRING (DLINE) EDIT
             ( TOTCIN, YECIN, NT, NFP )
             ( X(1), F(4), X(4), A(8), X(6), F(3), X(7), F(3) ) ;
        DISPLAY ( DLINE ) ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( '** LAST OF CIN DIRECTORY ENTRIES ENCOUNTERED **' ) ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( '** NOW STARTING CDP DIRECTORY **' ) ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( 'CDP NUM  CDP   REC PTR' ) ;
        DISPLAY ( '-------  ----  -------' ) ;
        DISPLAY ( '     ' ) ;

             CINPTR (1) = LOOP ;
             REFS(2) = LOOP ;
                  T35 = 2 ;
                  LOOP = 1 ;
             END LP3 ;

   /* PROCESS NEXT CDP ENTRY HERE */

        TOTCDP = TOTCDP + 1 ;
        LOOP = LOOP + 1 ;

             IF ( LOOP > 1200 ) THEN GO TO BUMCDP ;

                  CHCDP(LOOP) = OVERC.F1 ;
                  CDPPTR(LOOP) = OVERC.F2 ;

 PAGE NUMBER :    5

 SOURCE CARD LISTING FOR
  P7:      /* ABBREVIATED COURSE FILE DIRECTORY CONSTRUCT - 12/14/76 */

        PUT STRING (DLINE) EDIT
             ( TOTCDP, OVERC.F1, OVERC.F2 )
             ( X(1), F(4), X(4), A(4), X(4), F(4) ) ;
        DISPLAY ( DLINE ) ;

                        GO TO ANOTHER ;  /* ONTO NEXT SORT RECORD */

   /* HERE, OVERFLOWS TO DIRECTORY MEMORY TABLES */

BUMCIN: DISPLAY ( '     ' ) ;
        DISPLAY ( '     **** CIN TABLE OVERFLOW ****' ) ;
        NER = 1 ;
             GO TO BUM2 ;

BUMCDP: DISPLAY ( '     ' ) ;
        DISPLAY ( '     **** CDP TABLE OVERFLOW ****' ) ;
        NER = 2 ;

BUM2:   DISPLAY ( '     ' ) ;
        DISPLAY ( 'DIRECTORY CONSTRUCT PROGRAM IS TERMINATING'
             || ' DUE TO ERROR' ) ;

        CALL IHESARC ( 8 ) ;      /* STOP SORT RECORDS, EVEN IF MORE */
             GO TO ENDE35A ;

ENDE35A:     END E35A ;

/****************************************/

   /* HERE, BACK (ONE WAY OR ANOTHER) FROM THE SORT;
      EITHER THE SORT COMPLETED OK, OR ERROR FORCED
      THE TERMINATION */

        IF ( RCODE = 16 )
             THEN DISPLAY ( '*** SORT FAILED, CODE = 16 **' ) ;

             ELSE
             IF ( RCODE = 0 )
                  THEN DISPLAY ( '** SORT COMPLETED OK, CODE = 0 **');

                  ELSE
                  DO ;
                  PUT STRING (DLINE) EDIT
                        ( '** SORT RETURN CODE INVALID, ',
                         'CODE WAS ', RCODE, ' **' )
                        ( A, A, F(2), A ) ;
                  DISPLAY ( DLINE ) ;
                  END ;

   /* NOW, FINISH UP BY WRITING DIRECTORIES OUT TO DISK */

        IF ( RCODE ¬= 0 ) | ( NER > 0 ) THEN GO TO STEP99 ;

        CDPPTR (1) = LOOP ;

LP4:         DO J = 1 TO 4 ;
             PLCIN = ADDR ( COVCIN (J) ) ;
             WRITE FILE (DCIN) FROM (LOADCIN) ;
             END LP4 ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( '** CIN DIRECTORY HAS BEEN SUCCESSFULLY '
             || 'REWRITTEN' ) ;

LP5:         DO J = 1 TO 6 ;
             PLCDP = ADDR ( COVCDP (J) ) ;
             WRITE FILE (DCDP) FROM (LOADCDP) ;
             END LP5 ;
 PAGE NUMBER :    6

 SOURCE CARD LISTING FOR
  P7:       /* ABBREVIATED COURSE FILE DIRECTORY CONSTRUCT - 12/14/76 */

        DISPLAY ( '     ' ) ;
        DISPLAY ( '** CDP DIRECTORY HAS BEEN SUCCESSFULLY '
             || 'REWRITTEN' ) ;

STEP99: CLOSE FILE (DCIN), FILE (DCDP), FILE (COURSE) ;

        REWRITE FILE (MREF) FROM (MAST) ;
        CLOSE FILE (MREF) ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( 'COURSE DIRECTORY CONSTRUCT PROGRAM '
             || 'IS NOW TERMINATING' ) ;
        DISPLAY ( '     ' ) ;

             END P7 ;
```

```
PAGE NUMBER :   1

SOURCE CARD LISTING FOR
 P7A:    /* COURSE FILE DESCRIPTORS LOAD - WED/15/DEC/76 */
```

```
P7A:    /* COURSE FILE DESCRIPTORS LOAD - WED/15/DEC/76 */
        PROC OPTIONS (MAIN) ;

   /*   REVISIONS :
        WED/15/DEC/76 - FIRST IMPLEMENTATION.

        MON/14/MAR/77 - MODS TO PREPARE FOR SHIP TO NCSS.
   */

DECLARE 1 CINDESC      STATIC,
            2 CINDS    CHAR (8),
            2 FILL     CHAR (5),
            2 DTAB (100) FIXED BIN ;

DECLARE 1 CINCARD      STATIC,
            2 CINID    CHAR (8),
            2 FILL     CHAR (1),
            2 CINNUMS  CHAR (71),

        DLINE          CHAR (80) STATIC,
        CARD           CHAR (80) BASED (PCARD) ;

DECLARE YECIN          CHAR (8)
             STATIC INIT ( 'ABCDEFGH' ),
        GNUMS (24)     FIXED BIN STATIC,
        LCARD          CHAR (80) BASED (PLOAD),
        ICOURSE (160)  FIXED BIN STATIC,
        ( N15, N35, NT, NC, TOTCIN, T15, NS, J, CINTOT,
             NER )     FIXED BIN STATIC,
        RSORT          CHAR (213) STATIC ;

DECLARE DCARD     FILE RECORD INPUT,
        DESC      FILE RECORD OUTPUT,
        DIND      FILE RECORD INPUT ;

DECLARE 1 CINDER (304) STATIC,
            2 CHCIN    CHAR (8),
            2 CINPTR   FIXED BIN,
        LOADCIN        CHAR (760)  BASED (PLCIN),
        COVCIN (4)     CHAR (760)  BASED (PTCIN) ;

DECLARE 1 DINDEX       BASED (PIN),
            2 DSTART   FIXED BIN,
            2 DEND     FIXED BIN,
            2 NCAT     FIXED BIN,
            2 NDESC    FIXED BIN,
            2 CATEGORIES (15),
               3 PCAT  FIXED BIN,
               3 SCAT  FIXED BIN,
               3 ECAT  FIXED BIN,
            2 PDESC (111) FIXED BIN,
        COVER (4)      CHAR (80) BASED (PCOV) ;

DECLARE I MAST         STATIC,
            2 REFS (40)  FIXED BIN ;

/***************************************/

START:  NC, NS, TOTCIN, NEOF, LOOP,
             J, CINTOT, NT = 0 ;

        PIN = ADDR (ICOURSE) ;
        PCARD = ADDR (CINCARD) ;
        PTCIN = ADDR (CINDER) ;
        PCOV = ADDR (ICOURSE) ;
        PLOAD = ADDR ( COVER(1) ) ;

        OPEN FILE (DCARD) RECORD INPUT,
 PAGE NUMBER :   2

 SOURCE CARD LISTING FOR
  P7A:    /* COURSE FILE DESCRIPTORS LOAD - WED/15/DEC/76 */
             FILE (DIND) RECORD INPUT ;

        READ FILE (DIND) IGNORE (1) ;

LP1:         DO J = 1 TO 4 ;
             PLOAD = ADDR ( COVER (J) ) ;
             READ FILE (DIND) INTO (LCARD) ;
             END LP1 ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( 'COURSE FILE DESCRIPTOR LOAD PROGRAM' ) ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( '    CIN    ASSIGNED DESCRIPTOR(S)' ) ;
        DISPLAY ( '-------- ----------------------' ) ;
        DISPLAY ( '     ' ) ;

        CINDESC.FILL = ' ' ;

        OPEN FILE (MREF) RECORD UPDATE ;
        READ FILE (MREF) INTO (MAST) ;

        ON ENDFILE (DCARD) GO TO EOJ ;

        READ FILE (DCARD) INTO (CARD) ;
             NC = 1 ;

        DISPLAY ( CARD ) ;

                  GO TO STEP2 ;

/****************************************/

   /* NEXT DESCRIPTOR CARD INPUT HERE */

STEP1:  READ FILE (DCARD) INTO (CARD) ;
        NC = NC + 1 ;

        DISPLAY ( CARD ) ;

        IF ( CINID = YECIN ) THEN GO TO STEP3 ;

   /* HERE, STARTING A BRAND-NEW CIN; FIRST WRITE OLD-CIN
      RECORD OUT TO DESCRIPTOR FILE */

        WRITE FILE (DESC) FROM (CINDESC) ;
        NS = NS + 1 ;

STEP2:  YECIN = CINID ;                  /* STARTING NEW CIN HERE */
        TOTCIN = TOTCIN + 1 ;

STEP2B: CINDS = CINID ;                  /* CINS HERE ARE (=), START */
        DTAB = 0 ;                       /* MAKEP OF NEW DESC RECORD */

STEP3:  GNUMS = 0 ;                           /* EXTRACT SUBSCRIPTS */
        GET STRING (CINNUMS) EDIT
             (GNUMS) ( 24 ( F(2), X(1) ) ) ;

             NT = 0 ;
LP3:         DO J = 1 TO 24 ;            /* SUBSCRIPTS VERIFIED HERE */
             NTEST = GNUMS (J) ;
             IF ( NTEST = 0 ) THEN GO TO ENDLP3 ;
                  NT = NT + 1 ;
             IF ( PDESC ( NTEST ) ¬= 0 ) THEN GO TO STEP4 ;

        NER = NER + 1 ;

        DISPLAY ( '     ' ) ;
        PUT STRING (DLINE) EDIT
```

```
PAGE NUMBER :    3

SOURCE CARD LISTING FOR
 P7A:     /* COURSE FILE DESCRIPTORS LOAD - WED/15/DEC/76 */

             ( '** ERROR IN CIN -', CINID,
              '- DESCRIPTOR SUBSCRIPT ', NTEST,
              ' DOES NOT EXIST **' ) ( A, A, A, F(4), A ) ;
        DISPLAY ( DLINE ) ;
        DISPLAY ( '    ' ) ;

                  GO TO ENDLP3 ;

STEP4:       DTAB (NTEST) = PDESC (NTEST) ;
ENDLP3:      END LP3 ;

                  GO TO STEP1 ;

/**************************************/

   /* HERE, EOF ON INPUT DESCRIPTOR DECK */

EOJ:    WRITE FILE (DESC) FROM (CINDESC) ;      /* LAST RECORD */
        NS = NS + 1 ;

        REFS (5) = NC ;
        REFS (6) = NS ;
        REWRITE FILE (MREF) FROM (MAST) ;

EOJ2:   CLOSE FILE (DESC), FILE (DIND),
              FILE (MREF), FILE (DCARD) ;

        DISPLAY ( '    ' ) ;
        DISPLAY ( '** NOW AT EOF ON DESCRIPTOR CARD INPUT **' ) ;

        DISPLAY ( '    ' ) ;
        PUT STRING (DLINE) EDIT
             ( '    TOTAL NUMBER DESCRIPTOR INPUT CARDS : ', NC )
             ( A, P'Z,ZZ9' ) ;
        DISPLAY ( DLINE ) ;

        PUT STRING (DLINE) EDIT
             ( '      TOTAL UNIQUE CIN NUMBERS IN INPUT : ', TOTCIN )
             ( A, P'Z,ZZ9' ) ;
        DISPLAY ( DLINE ) ;

        PUT STRING (DLINE) EDIT
             ( 'TOTAL NUMBER OUTPUT DESCRIPTOR RECORDS : ', NS )
             ( A, P'Z,ZZ9' ) ;
        DISPLAY ( DLINE ) ;

        PUT STRING (DLINE) EDIT
             ( '  TOTAL NUMBER CIN OR SUBSCRIPT ERRORS : ', NER )
             ( A, P'Z,ZZ9' ) ;
        DISPLAY ( DLINE ) ;

        DISPLAY ( '    ' ) ;
        DISPLAY ( 'COURSE DESCRIPTOR LOAD PROGRAM IS '
             || 'TERMINATING' ) ;
        DISPLAY ( '    ' ) ;

             END P7A ;
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
 P7B:    /* ABBREVIATED COURSE FILE LISTINGS - MON/17/JAN/77 */

P7B:    /* ABBREVIATED COURSE FILE LISTINGS - MON/17/JAN/77 */
        PROC (INPARM) OPTIONS (MAIN) ;

   /*   REVISIONS :
        MON/17/JAN/77 - FIRST IMPLEMENTATION.

        SUN/20/MAR/77 - MODS PRIOR TO INSTALLATION AT NCSS.
   */

DECLARE INPARM    CHAR (100) VARYING,
        1 POVER   BASED (PP),
          2 PARM  CHAR (1),
          2 PARM2 CHAR (4) ;

DECLARE INDESC         FILE RECORD DIRECT
             KEYED ENV ( REGIONAL(1) ),

        COURSE         FILE RECORD DIRECT INPUT
             KEYED ENV ( REGIONAL(1) ) ;

DECLARE T1        CHAR (74) STATIC INIT ( '   CIN   CDP    COURSE TITLE
   NOBC NEC  RMS  TPC  LEN THRS LHRS ATTR STBX' ),

        T2        CHAR (74) STATIC INIT ( '-------- ---- -------------
-- ---- ---- --- ----- --- ---- ---- ---- ----' ),

        1 T3      STATIC,
          2 F1    CHAR (15) INIT ( '    PRIORITY : ' ),
          2 PC    CHAR (2),
          2 F2    CHAR (16) INIT ( '     TYPE CRS : ' ),
          2 TYCRS CHAR (2),
          2 F3    CHAR (19) INIT ( '     SERVICE CDE : ' ),
          2 SVC   CHAR (1),
          2 F4    CHAR (16) INIT ( '     COST/AOB : ' ),
          2 CAOB  PICTURE '(4)Z9V.99',

        T3OVER    CHAR (79) BASED (PT3),

        1 T4      STATIC,
          2 F1    CHAR (15) INIT ( '       TRAPS : ' ),
          2 TRAPS CHAR (1),
          2 F2    CHAR (17) INIT ( '        STATUS : ' ),
          2 STCD  CHAR (1),
          2 F3    CHAR (20) INIT ( '       STATUS DTE : ' ),
          2 STDTE PICTURE '(5)9',
          2 F4    CHAR (12) INIT ( '   METH-I : ' ),
          2 MI    CHAR (1),

        T4OVER    CHAR (72) BASED (PT4),

        D2             CHAR (2) DEF DLINE POS (1),
        DLINE     CHAR (80) STATIC ;

DECLARE ELINE     CHAR (38) STATIC INIT (
             '** ERROR XX OCCURED; NREC WAS XXXXX **' ),
        END       CHAR (2) DEF ELINE POS (10),
        ENREC     PICTURE 'ZZZZ9' DEF ELINE POS (31),

        ( DISP, PTR, TPAGE, J, JKEY, LL, LP, LT, MAX, MAXC,
             MAXCDP, MAXCKEY, MAXLD, MAXLP, NREC, NSK,
             PPAGE )   FIXED BIN STATIC,

        ( RTA(3), RTB(2),
             RTC(2) )  LABEL ;

/***************************************/

DECLARE 1 CDPDIR (1200) STATIC,       /* CRS CDP DIRECTORY */
```

```
 PAGE NUMBER :    2

 SOURCE CARD LISTING FOR
  P7B:    /* ABBREVIATED COURSE FILE LISTINGS - MON/17/JAN/77 */
          2 CHCDP       CHAR (4),       /* CKF.CRS.DIRCDP */
          2 CDPPTR      FIXED BIN,
        LOADCDP         CHAR (1200) BASED (PLCDP),
        COVCDP (6)      CHAR (1200) BASED (PTCDP) ;

DECLARE 1 DINDEX        STATIC,         /* OVERALL DESCRIPTOR INDEX */
          2 DSTART      FIXED BIN,      /* CKF.DESC.INDEX */
          2 DEND        FIXED BIN,
          2 NCAT        FIXED BIN,
          2 NDESC       FIXED BIN,
          2 CATEGORIES (15),
            3 PCAT      FIXED BIN,
            3 SCAT      FIXED BIN,
            3 ECAT      FIXED BIN,
          2 PDESC (111) FIXED BIN,

        COVER (4)       CHAR (80) BASED (PCOV) ;

DECLARE 1 MAST          STATIC,
          2 REFS (40)   FIXED BIN,
        CARD            CHAR (80) STATIC ;

DECLARE 1 ACRS          STATIC,         /* ABBR. COURSE FILE */
          2 SET1,                       /* CKF.ACOURSES */
            3 CDP       CHAR (4),
            3 CIN       CHAR (8),
            3 CST       CHAR (16),
            3 NOBC      CHAR (4),
            3 NEC       CHAR (4),
          2 PC          CHAR (2),
          2 SET2,
            3 RMS       CHAR (3),
            3 TYCRS     CHAR (2),
            3 SVC       CHAR (1),
            3 MI        CHAR (1),
          2 STCD        CHAR (1),
          2 STDTE       PICTURE '(5)9',
          2 SET3,
            3 TRAPS     CHAR (1),
            3 TPC       CHAR (5),
          2 ATTR        PICTURE '99V9',
          2 STBK        PICTURE '99V9',
          2 CLEN        PICTURE '999',
          2 THRS        PICTURE '999',
          2 LHRS        PICTURE '999',
          2 CADB        PICTURE '(5)9V99',
          2 CCBO        CHAR (1) ;

/**************************************/

START:  PT3 = ADDR (T3) ;
        PT4 = ADDR (T4) ;
        PP = ADDR (INPARM) ;
        PTCDP = ADDR (CDPDIR) ;
        PCOV = ADDR (DINDEX) ;

        TPAGE, PPAGE = 0 ;
        MAXLP = 58 ;
        MAXLD = 60 ;
        NREC = -1 ;
        JKEY = 1 ;
        LL = 1 ;

COMMENCE:
        OPEN FILE (MREF) RECORD INPUT ;
        READ FILE (MREF) INTO (MAST) ;

             MAXCDP = REFS(1) ;
             MAXCKEY = REFS(1) + 2 ;
```

```
PAGE NUMBER :    3

SOURCE CARD LISTING FOR
 P78:    /* ABBREVIATED COURSE FILE LISTINGS - MON/17/JAN/77 */

             MAXC = REFS(4) ;

        IF ( PARM = 'C' ) THEN GO TO CSETUP ;

        OPEN FILE (COURSE) DIRECT INPUT ;

        IF ( PARM = 'A' ) THEN GO TO FIRST ;

BSETUP: OPEN FILE (DCDP) RECORD INPUT ;
CDPLOAD:     DO J = 1 TO 6 ;
            PLCDP = ADDR ( CDVCDP(J) ) ;
             READ FILE (DCDP) INTO (LOADCDP) ;
             END CDPLOAD ;

CSETUP: OPEN FILE (INDESC) INPUT ;
             DO J = 1 TO 4 ;        /* NOTE, SKIPS RECORD 0 */
             READ FILE (INDESC) INTO (CARD) KEY (J) ;
             COVER (J) = CARD ;
             END ;

        NREC = DSTART - 1 ;

FIRST:  DISP, PTR = 1 ;
        IF ( PARM2 = 'BOTH' ) THEN GO TO STEP1 ;

        IF ( PARM2 = 'TERM' )
             THEN PTR = 2 ;
             ELSE DISP = 2 ;

STEP1:  LT, LP = 100 ;
        IF ( PTR = 2 ) THEN GO TO STEP6 ;

   /* HERE, PRINTER DATASET IS INITILIZED */

        OPEN FILE (MPTR) STREAM PRINT
             PAGESIZE (58) LINESIZE (120) ;

        ON ENDPAGE (MPTR)
             BEGIN ;
             PPAGE = PPAGE + 1 ;

             IF ( PARM = 'C' ) THEN GO TO STEP2 ;

             PUT FILE (MPTR) EDIT
                  ( 'PAGE NO : ', PPAGE, T1, T2, '     ' )
                  ( PAGE, LINE(2), X(10), A, F(4),
                    2 ( SKIP(1), X(10), A(74) ), SKIP(1), A ) ;
                  GO TO SKP ;

STEP2:       PUT FILE (MPTR) EDIT ( 'PAGE NO : ', PPAGE,
                  '*** LISTING OF DESCRIPTOR CATEGORIES AS',
                  '     APPLIED TO COURSE DATA ***', '     ' )
                  ( PAGE, LINE(5), X(20), A, F(2),
                    SKIP(3), X(20), A, SKIP(1), X(20), A,
                    SKIP(3), A ) ;

SKP:         PUT FILE (MPTR) SKIP (1) ;
STEP3:       END ;

        SIGNAL ENDPAGE (MPTR) ;

STEP6:  IF ( PARM ¬= 'C' ) THEN GO TO PLOOP ;

   /* PRINTING LOOP FOR C-OPTION FOLLOWS */

CLOOP:  NREC = NREC + 1 ;
ER2:    IF ( NREC > MAXC ) THEN GO TO BUST2 ;
        READ FILE (INDESC) INTO (DLINE) KEY (NREC) ;
             GO TO ROUTEB ;
```

```
SOURCE CARD LISTING FOR
 P7B:    /* ABBREVIATED COURSE FILE LISTINGS - MON/17/JAN/77 */

   /* PRINTING LOOP FOR A-B OPTIONS */

PLOOP:  LL = 1 ;
        IF ( PARM = 'B' ) THEN GO TO BREAD ;

AREAD:  NREC = NREC + 1 ;
        IF ( NREC > MAXCDP ) THEN GO TO EOFA[illegible]
GETCRS: READ FILE (COURSE) INTO (ACRS) KEY (N[illegible]) ;
             GO TO ROUTEA ;

BREAD:  JKEY = JKEY + 1 ;
        IF ( JKEY > MAXCKEY ) THEN GO TO EOFAB
        NREC = CDPPTR (JKEY) ;
ER1:    IF ( NREC > MAXCDP ) THEN GO TO BUST1 ;
             GO TO GETCRS ;

   /* ROUTING DONE HERE BASED ON PRINT LINE NO 1,2, OR 3 */

ROUTEA: GO TO RTA (LL) ;

RTA(1): PUT STRING (DLINE) EDIT
             ( CIN, CDP, CST, NOBC, NEC, RMS, TPC,
               CLEN, THRS, LHRS, ATTR, STBK )
             ( A(8), X(1), A(4), X(1), A(16), X(1), A(4), X(1),
               A(4), X(1), A(3), X(1), A(5), X(1), P'ZZ9',
               2 ( X(2), P'ZZ9' ), 2 ( X(1), P'Z9V.9' ) ) ;
                  GO TO ROUTEB ;

RTA(2): T3 = ACRS, BY NAME ;
        T3.TYCRS = ACRS.SET2.TYCRS ;
        T3.SVC = ACRS.SET2.SVC ;
        DLINE = T3OVER ;
             GO TO ROUTEB ;

RTA(3): T4 = ACRS, BY NAME ;
        T4.TRAPS = ACRS.SET3.TRAPS ;
        T4.MI = ACRS.SET2.MI ;
        DLINE = T4OVER ;

   /* NOW, SETUP FOR OUTPUT OF A NEW LINE */

ROUTEB:      GO TO RTB (DISP) ;

RTB(1): /* HERE TERMINAL IS SERVICED */
        LT = LT + 1 ;

        IF ( PARM = 'C' ) THEN GO TO LP2A ;

        IF ( LL = 1 ) THEN
LP1:         DO ;
             LT = LT + 1 ;
             DISPLAY ( '  ' ) ;
             END LP1 ;

LP2A:   IF ( LT > MAXLD ) THEN
LP2:         DO ;
             MAX = MAXLD + 4 ;

LP3:              DO J = LT TO MAX ;
                  DISPLAY ( '  ' ) ;
                  END LP3 ;

             TPAGE = TPAGE + 1 ;
             DISPLAY ( '    PAGE : ' || TPAGE ) ;
             DISPLAY ( '  ' ) ;
             LT = 4 ;

             IF ( PARM = 'C' ) THEN GO TO LP3A ;
 PAGE NUMBER :    5

 SOURCE CARD LISTING FOR
  P7B:     /* ABBREVIATED COURSE FILE LISTINGS - MON/17/JAN/77 */

             DISPLAY ( T1 ) ;
             DISPLAY ( T2 ) ;
             DISPLAY ( '  ' ) ;

             LT = 7 ;
LP3A:        END LP2 ;

        IF ( PARM = 'C') & ( (D2 = '**') | (D2 = '--') )
             THEN DO ;  DISPLAY ( '     ' ) ;  LT = LT + 1 ;  END ;

        DISPLAY ( DLINE ) ;

RTB(2): /* IF REQUIRED, OFFLINE PTR SERVICED HERE */
        GO TO RTC ( PTR ) ;

RTC(1): /* HERE, LINE PRINTER IS SERVICED */
        NSK = 1 ;
        IF ( PARM = 'C' ) THEN GO TO LINEOUT ;

        IF ( LL = 1 ) & ( LINENO(MPTR) > 54 )
             THEN SIGNAL ENDPAGE (MPTR) ;

        IF ( LL = 1 ) THEN NSK = 2 ;

LINEOUT:
        IF ( PARM = 'C') & ( (D2 = '**') | (D2 = '--') )
             THEN NSK = 2 ;

        PUT FILE (MPTR) EDIT (DLINE)
             ( SKIP(NSK), X(10), A(80) ) ;

   /* NOW, REGROUP FOR NEXT LINE, IF ANY */

RTC(2):
TALLY:  IF ( PARM ¬= 'C' ) THEN GO TO STEP5 ;

        IF ( NREC >= DEND )
             THEN GO TO EOFAB ;
             ELSE GO TO CLOOP ;

STEP5:  LL = LL + 1 ;
        IF ( LL < 4 ) THEN GO TO ROUTEA ;
                      ELSE GO TO PLOOP ;

/****************************************/

   /* MAX KEY EXCEEDED ERRORS PROCESSED HERE */

BUST1:  ENO = '01' ;
             GO TO STEP4 ;
BUST2:  ENO = '02' ;

STEP4:  ENREC = NREC ;

        IF ( DISP = 1 ) THEN
LP4:         DO ;
             DISPLAY ( '    ' ) ;
             DISPLAY ( ELINE )  ;
             DISPLAY ( '    ' ) ;
             END LP4 ;

        IF ( PTR = 1 ) THEN
             PUT FILE (MPTR) EDIT (ELINE)
            ( SKIP(2), X(20), A(38) ) ;

/****************************************/

   /* HERE, EOF OCCURED ON INPUT FILE */
```

```
PAGE NUMBER :    6

SOURCE CARD LISTING FOR
 P7B:     /* ABBREVIATED COURSE FILE LISTINGS - MON/17/JAN/77 */


EOFAB:  IF ( PTR = 1 )
             THEN CLOSE FILE (MPTR) ;

             END P7B ;
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
 P8:       /* VEHICLE DATA/DESCRIPTOR LOAD, DIRECTORY - MON/24/JAN/77 */

P8:        /* VEHICLE DATA/DESCRIPTOR LOAD, DIRECTORY - MON/24/JAN/77 */
           PROC OPTIONS (MAIN) ;

   /*      REVISIJNS :
           MON/24/JAN/77 - INITIAL IMPLEMENTATION.

           TUE/15/MAR/77 - MODS PRIOR TO SHIPPING TO NCSS..
   */

DECLARE CARD             CHAR (80) STATIC,
        DLINE            CHAR (80) STATIC,
        CC7J79           CHAR (10) DEF CARD POS (70),
        FVEHS (200)      FIXED BIN STATIC,
        NODATA (200)     CHAR (13) STATIC,
        ( D1FLAG, D2FLAG, ND, NV, NT, K, NSK,
             NTEST, TER, NER, NDD, NFIND, NOD, TOTD, NS,
             NDIR, J ) FIXED BIN STATIC ;

DECLARE 1 DINDEX         STATIC,            /* OVERALL DESCRIPTOR INDEX */
          2 DSTART       FIXED BIN,         /* CKF.DESC.INDEX */
          2 DEND         FIXED BIN,
          2 NCAT         FIXED BIN,
          2 NDESC        FIXED BIN,
          2 CATEGORIES (15),
            3 PCAT       FIXED BIN,
            3 SCAT       FIXED BIN,
            3 ECAT       FIXED BIN,
          2 PDESC (111) FIXED BIN,
        COVER (4)        CHAR (80) BASED (PCOV) ;

DECLARE 1 MAST           STATIC,
          2 REFS (40)    FIXED BIN,

        DIND             FILE RECORD DIRECT
             KEYED ENV ( REGIONAL(1) ) ;

/****************************************/

DECLARE 1 VEHCARD        STATIC,            /* VEH DESCRIPTOR CARD */
          2 VSN          CHAR (13),         /* CKF.VDESC */
          2 FILL         CHAR (1),
          2 VEHNUMS      CHAR (66),

        VEH80            CHAR (80) BASED (PVV80),
        GVNS (22)        FIXED BIN STATIC,
        YEVEH            CHAR (13) STATIC INIT ( 'ABCDEFGH' ),

        1 VEHDESC        STATIC,            /* VEH DESCRIPTOR FILE */
          2 VEHDS        CHAR (13),         /* CKF.VEH.DESC */
          2 VTAB (100) FIXED BIN ;

DECLARE 1 VEHDIR (200) STATIC,              /* VEHICLE DIRECTORY */
          2 CHVEH        CHAR (13),         /* CKF.VEH.DIR */
          2 VEHPTR       FIXED BIN,

        COVVEH (4)       CHAR (750) BASED (PTVEH),
        LOADVEH          CHAR (750) BASED (PLVEH) ;

DECLARE 1 AVEHICLES      STATIC,            /* ABBR. VEH FILE */
          2 STJCKN       CHAR (13),         /* CKF.AVEHS */
          2 DEVDESIG     CHAR (9),
          2 DEVNAME      CHAR (47),
          2 DEVCOST      PICTURE '(8)9V99',
          2 CC80         CHAR (1),

        VCOST            FIXED DEC (10,2) STATIC,
        VOVER            CHAR (80) BASED (PV80) ;
```

```
PAGE NUMBER :   2

SOURCE CARD LISTING FOR
 P8:      /* VEHICLE DATA/DESCRIPTOR LOAD, DIRECTORY - MON/24/JAN/77 */
/****************************************/

START:  ND, NV, TER, NER, NS,
             TOTD, NFIND, NOC,
             J = 0 ;

        PCOV = ADDR (DINDEX) ;
        PVV80 = ADDR (VEHCARD) ;
        PV80 = ADDR (AVEHICLES) ;
        PTVEH = ADDR (VEHDIR) ;

        NDIR = -1 ;              /* DIRECTORY RECORD POINTER */
        NDD = 1 ;                /* DIRECTORY LOAD SS, STARTS AT 2 */
        FVEHS = 0 ;              /* VEH NUMBER FLAG TABLE */
        CHVEH = ' ' ;            /* VEH DIRECTORY */
        VEHPTR = 0 ;

        OPEN FILE (MREF) RECORD UPDATE,
             FILE (DIND) INPUT,
             FILE (VDATA) RECORD INPUT,
             FILE (VCDESC) RECORD INPUT,
             FILE (DVEH) RECORD OUTPUT,
             FILE (VDESC) RECORD OUTPUT,
             FILE (VEHS) RECORD OUTPUT ;

        ON ENDFILE (VCDESC) GO TO EOFDESC ;
        ON ENDFILE (VDATA) GO TO EOFDATA ;

        READ FILE (MREF) INTO (MAST) ;

LP1:         DO J = 5 TO 8 ;
             READ FILE (DIND) INTO (CARD) KEY (J) ;
             COVER ( J - 4 ) = CARD ;
             END LP1 ;

/****************************************/

    /* VEHICLE FILE DATA INPUT PROCESSED HERE */

        DISPLAY ( '     ' ) ;
        DISPLAY ( 'INITILIZE OF VEHICLE DATA AND DESCRIPTOR FILES' ) ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( 'NUM  STOCK NUMBER   DESIGNATOR  DEVICE '
             || 'NOMENCLATURE' ) ;
        DISPLAY ( '---  ------------   ----------  -------'
             || '------------' ) ;
        DISPLAY ( '     ' ) ;

STEP1:  READ FILE (VDATA) INTO (CARD) ;
        NV = NV + 1 ;

        GET STRING (CC7079) EDIT (VCOST) ( F(10,2) ) ;
        PUT STRING (CC7079) EDIT (VCOST) ( P'(8)9V99' ) ;

             VJVER = CARD ;

        PUT STRING (DLINE) EDIT
             ( NV, STOCKN, DEVDESIG, DEVNAME )
             ( F(3), X(2), A, X(2), A, X(2), A ) ;
        DISPLAY ( DLINE ) ;

        IF ( STOCKN = YEVEH ) THEN
DUPS:        DO ;
             NV = NV - 1 ;

             DISPLAY ( '     ' ) ;
        DISPLAY ( '      ** DUPLICATE STOCK NUMBER ON PRECEEDING CARD');
        DISPLAY ( '         RECORD WILL BE IGNORED **' ) ;
 PAGE NUMBER :   3

 SOURCE CARD LISTING FOR
  P8:       /* VEHICLE DATA/DESCRIPTOR LOAD, DIRECTORY - MON/24/JAN/77 */
             DISPLAY ( '     ' ) ;

                  GO TO STEP1 ;
             END DUPS ;

        WRITE FILE (VEHS) FROM (CARD) ;

        YEVEH = STOCKN ;
        NDD = NDD + 1 ;
        NDIR = NDIR + 1 ;
        VEHPTR (NDD) = NDIR ;
        CHVEH (NDD) = STOCKN ;

                  GO TO STEP1 ;

   /* HERE, EOF ON CARD DATA INPUT DATASET */

EOFDATA: VEHPTR (1) = NDD ;  /* LAST SS USED (1-ORGIN) IN DIR */

             DISPLAY ( '     ' ) ;
        DISPLAY ( '** NOW AT EOF ON DATA CARD INPUT **' ) ;

        PUT STRING (DLINE) EDIT
             ('    A TOTAL OF ', NV, ' INPUT CARDS WERE READ' )
             ( A, P'Z,ZZ9', A ) ;
        DISPLAY ( DLINE ) ;

LP2:         DO J = 1 TO 4 ;      /* WRITE OUT DIRECTORY */
             PLVEH = ADDR ( COVVEH(J) ) ;
             WRITE FILE (DVEH) FROM (LOADVEH) ;
             END LP2 ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( '** VEHICLE DIRECTORY HAS BEEN SUCCESSFULLY '
             || 'REWRITTEN' ) ;

        REFS (7) = NV - 1 ;
        REFS (8) = NDD ;

/***************************************/

   /* INITILIZE START OF DESCRIPTOR LOAD HERE */

        DISPLAY ( '     ' ) ;
      DISPLAY ( '** LOAD OF VEHICLE DESCRIPTOR FILE FOLLOWS **' ) ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( 'STOCK NUMBER   DEVICE DESCRIPTORS' ) ;
        DISPLAY ( '------------  ------------------' ) ;
        DISPLAY ( '     ' ) ;

        READ FILE (VCDESC) INTO (VEHCARD) ;
        ND = 1 ;
             GO TO STEP3 ;

   /* HERE IS THE DESCRIPTOR LOAD LOOP */

STEP2:  READ FILE (VCDESC) INTO (VEHCARD) ;
        ND = ND + 1 ;

        IF ( VSN = YEVEH ) THEN
LP3:         DO ;
             NSK = 1 ;
                  GO TO STEP4 ;
             END LP3 ;

        WRITE FILE (VDESC) FROM (VEHDESC) ;
        NS = NS + 1 ;
 PAGE NUMBER :   4

 SOURCE CARD LISTING FOR
  P8:      /* VEHICLE DATA/DESCRIPTOR LOAD, DIRECTORY - MON/24/JAN/77 */
STEP3:  YEVEH = VSN ;
        TOTD = TOTD + 1 ;
        VTAB = 0 ;
        VEHDS = VSN ;
             NSK = 2 ;

LP5:         DO J = 1 TO NDD ;    /* SEE IF MATCHING DATA AVAILABLE */
             IF ( VSN ¬= CHVEH(J) ) THEN GO TO LP5END ;
                  NFIND = NFIND + 1 ;
                  FVEHS (J) = 1 ;
                       GO TO STEP4 ;
LP5END:      END LP5 ;

        NOD = NOD + 1 ;
        NODATA (NOD) = VSN ;

STEP4:  DLINE = VSN || '    ' || VEHNUMS ;
        DISPLAY ( DLINE ) ;

        GVNS = 0 ;
        GET STRING (VEHNUMS) EDIT (GVNS) ( 22 ( F(2), X(1) ) ) ;

             NT = 0 ;
LP4:         DO J = 1 TO 22 ;
             NTEST = GVNS (J) ;
             IF ( NTEST = 0 ) THEN GO TO ENDLP4 ;
                  NT = NT + 1 ;
             IF ( PDESC(NTEST) ¬= 0 ) THEN GO TO STEP5 ;

        NER = NER + 1 ;
        DISPLAY ( '     ' ) ;
        PUT STRING (DLINE) EDIT
             ( '** ERROR IN VEH -', VSN, '- DESCRIPTOR SUBSCRIPT ',
               NTEST, ' DOES NOT EXIST **' )
             ( A, A, A, F(4), A ) ;
        DISPLAY ( DLINE ) ;

        IF ( NER > 100 ) THEN GO TO EOJ ;
             GO TO ENDLP4 ;

STEP5:       VTAB (NTEST) = PDESC (NTEST) ;
ENDLP4:      END LP4 ;

                  GO TO STEP2 ;

   /* HERE IS EOF ON DESCRIPTOR CARD INPUT */

EOFDESC: WRITE FILE (VDESC) FROM (VEHDESC) ;
         NS = NS + 1 ;

         REFS (9) = NS ;
         REFS (10) = ND ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( '** EOF REACHED ON DESCRIPTOR CARD INPUT FILE **' ) ;

        DISPLAY ( '     ' ) ;
        PUT STRING (DLINE) EDIT
             ( '          TOTAL NUMBER DESCRIPTOR CARDS INPUT : ',
               ND ) ( A, P'Z,ZZ9' ) ;
        DISPLAY ( DLINE ) ;

        PUT STRING (DLINE) EDIT
             ( '    TOTAL NUMBER UNIQUE VEHICLE STOCK NUMBERS : ',
               TOTD )  ( A, P'Z,ZZ9' ) ;
        DISPLAY ( DLINE ) ;

        PUT STRING (DLINE) EDIT
             ( 'TOTAL MATCHES FOUND IN VEHICLE DATA DIRECTORY : ',
 PAGE NUMBER :   5

 SOURCE CARD LISTING FOR
  P8:     /* VEHICLE DATA/DESCRIPTOR LOAD, DIRECTORY - MON/24/JAN/77 */

               NFIND )  ( A, P'Z,ZZ9' ) ;
        DISPLAY ( DLINE ) ;

/****************************************/

   /* ANY REQUIRED DIAGNOSTICS OUTPUT IN THIS SECTION */

         IF ( NFIND = NV ) THEN GO TO STEP6 ;

   /* HERE DESCRIPTOR / DATA MISMATCHES PRINTED */

        DISPLAY ( '     ' ) ;   DISPLAY ( '     ' ) ;
        DISPLAY ( '** THE FOLLOWING DATA FILE VEH NUMBERS' ) ;
        DISPLAY ( '    HAVE NO MATCH IN THE VEH DESCRIPTOR INPUT **' ) ;
        DISPLAY ( '     ' ) ;

             K = 0 ;
LP6:         DO J = 1 TO NOD ;
             IF ( FVEHS(J) = 1 ) THEN GO TO LP6END ;
             K = K + 1 ;

        DISPLAY ( (5)' ' || CHVEH(J) ) ;

LP6END:      END LP6 ;

   /* HERE, DESCRIPTOR VEH NUMS W/O MATCHING DATA FILE ENTRIES
      ARE PRINTED - IF ANY */

STEP6:  IF ( NOD = 0 ) THEN GO TO BITTEREND ;

        DISPLAY ( '     ' ) ;   DISPLAY ( '     ' ) ;
        DISPLAY ( '** THE FOLLOWING DESCRIPTOR CARD VEH NUMBERS' ) ;
        DISPLAY ( '    HAVE NO MATCH IN THE VEH DATA FILE **' ) ;
        DISPLAY ( '     ' ) ;

LP7:         DO J = 1 TO NOD ;
        DISPLAY ( (5)' ' || NODATA(J) ) ;
             END LP7 ;

/****************************************/

   /* HERE AM THE WINDUP */

BITTEREND:
        REWRITE FILE (MREF) FROM (MAST) ;

EOJ:    CLOSE FILE (MREF), FILE (VDATA), FILE (VCDESC),
             FILE (DIND), FILE (VDESC),
             FILE (DVEH), FILE (VEHS) ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( 'VEHICLE DATA/DIRECTORY/DESCRIPTOR LOAD PROGRAM' ) ;
        DISPLAY ( '       IS NOW TERMINATING' ) ;
        DISPLAY ( '     ' ) ;

             END P8 ;
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
 P8A:    /* ABBREVIATED VEHICLE FILE LISTINGS - WED/2/FEB/77 */

P8A:    /* ABBREVIATED VEHICLE FILE LISTINGS - WED/2/FEB/77 */
        PROC (INPARM) OPTIONS (MAIN) ;

   /*   REVISIONS :
        WED/02/FEB/77 - FIRST IMPLEMENTATION.

        SUN/20/MAR/77 - MODS PRIOR TO INSTALLATION AT NCSS.
   */

DECLARE ( DIND, VEHS ) FILE RECORD DIRECT
                   KEYED ENV ( REGIONAL(1) ),

        INPARM         CHAR (100) VARYING,
        1 POVER        BASED (PP),
          2 PARM       CHAR (1),
          2 PARM2      CHAR (4),

        VCOST          FIXED DEC (10,2) STATIC,
        DLINE          CHAR (80) STATIC,
        D2             CHAR (2) DEF DLINE POS (1),
        CARD           CHAR (80) STATIC,

        T1             CHAR (46) STATIC INIT (
'ITEM  STOCK-NUMBER   DESIGNATOR   VEHICLE COST' ),

        T2             CHAR (46) STATIC INIT (
'----  -------------  ----------   ------------' ),

        T3             CHAR (11) STATIC INIT (
'PAGE : XXXX' ),
        T3P            PICTURE 'ZZZ9' DEF T3 POS (8),

        ELINE          CHAR (39) STATIC INIT (
'** ERROR XX OCCURED ; NREC WAS XXXXX **' ),
        ENO            PICTURE '99' DEF ELINE POS (10),
        ENREC          PICTURE 'Z,ZZ9' DEF ELINE POS (32),

        ( DISP, PTR, NPAGE, DPAGE, LT, JKEY, L, LP, NC, NREC,
             MAXC, MAXIND, MAXLT, MAXLTA, MAXLTB, MAXSS, MAXVEH,
             J, K )    FIXED BIN STATIC ;

/*************************************/

DECLARE 1 DINDEX       STATIC,        /* OVERALL DESCRIPTOR INDEX */
          2 DSTART     FIXED BIN,     /* CKF.DESC.INDEX */
          2 DEND       FIXED BIN,
          2 NCAT       FIXED BIN,
          2 NDESC      FIXED BIN,
          2 CATEGORIES (15),
            3 PCAT     FIXED BIN,
            3 SCAT     FIXED BIN,
            3 ECAT     FIXED BIN,
          2 PDESC (111) FIXED BIN,
        COVER (4)      CHAR (80) BASED (PCOV) ;

DECLARE 1 MAST         STATIC,
          2 REFS (40)  FIXED BIN,
        MREF           FILE RECORD ;

DECLARE 1 VEHDIR (200) STATIC,        /* VEHICLE DIRECTORY */
          2 CHVEH      CHAR (13),     /* CKF.VEH.DIR */
          2 VEHPTR     FIXED BIN,
        COVVEH (4)     CHAR (750) BASED (PTVEH),
        LOADVEH        CHAR (750) BASED (PLVEH) ;

DECLARE 1 AVEHICLES    STATIC,        /* ABBR. VEH FILE */
          2 STOCKN     CHAR (13),     /* CKF.AVEHS */
          2 DEVDESIG   CHAR (9),
 PAGE NUMBER :    2

 SOURCE CARD LISTING FOR
  P8A:     /* ABBREVIATED VEHICLE FILE LISTINGS - WED/2/FEB/77 */

          2 DEVNAME    CHAR (47),
          2 DEVCOST    PICTURE '(8)9V99',
          2 CC80       CHAR (1),
        VOVER          CHAR (80) BASED (PV80) ;

/**************************************/

START:  NPAGE, DPAGE, NC,
             J, K = 0 ;

        LT = 100 ;          /* DISPLAY PAGE LINE COUNT */
        MAXLT = 60 ;        /* MAX # LINES ON DISPLAY */
        JKEY = 1 ;          /* SS IN DIRECTORY FOR EXTRACT -
                               NOTE - SKIPS FIRST LOC VEHPTR(1) */
        PP = ADDR (INPARM) ;

        PV80 = ADDR (AVEHICLES) ;
        PTVEH = ADDR (VEHDIR) ;
        PCOV = ADDR (DINDEX) ;

COMMENCE:
        OPEN FILE (MREF) RECORD INPUT,
             FILE (DIND) INPUT,
             FILE (VEHS) INPUT,
             FILE (DVEH) RECORD INPUT ;

        READ FILE (MREF) INTO (MAST) ;
        MAXVEH = REFS (7) ;
        MAXC = REFS (4) ;

        IF ( PARM = 'B' ) THEN GO TO BSETUP ;

ASETUP: MAXSS = REFS (8) ;  /* HERE, PLAN FOR ABBREVIATED DATA FILE */

DIRLOAD:     DO J = 1 TO 4 ;
             PLVEH = ADDR ( COVVEH(J) ) ;
             READ FILE (DVEH) INTO (LOADVEH) ;
             END DIRLOAD ;

        MAXLTA = MAXLT - 3 ;

                  GO TO FIRST ;

BSETUP:      DO J = 5 TO 8 ;     /* SETUP FOR DESCRIPTOR LIST */
             READ FILE (DIND) INTO (CARD) KEY (J) ;
             COVER ( J - 4 ) = CARD ;
             END BSETUP ;

        MAXIND = REFS (4) ;
        NREC = DSTART - 1 ;
        MAXLTB = MAXLT - 2 ;

/**************************************/

FIRST:  DISP, PTR = 1 ;     /* DEVICE ASSIGNMENTS CHECKED HERE */

        IF ( PARM2 = 'BOTH' ) THEN GO TO STEP1 ;

        IF ( PARM2 = 'TERM' )
             THEN PTR = 2 ;
             ELSE DISP = 2 ;

STEP1:  LT, LP = 100 ;

   /* HERE, PRINTER DATASET IS INITILIZED */

        IF ( PTR = 2 ) THEN GO TO STEP4 ;

        OPEN FILE (SYSPRINT)
 PAGE NUMBER :   3

 SOURCE CARD LISTING FOR
  P8A:      /* ABBREVIATED VEHICLE FILE LISTINGS - WED/2/FEB/77 */
             PAGESIZE (58) LINESIZE (120) ;

        ON ENDPAGE (SYSPRINT)
             BEGIN ;
             NPAGE = NPAGE + 1 ;

             IF ( PARM = 'B' ) THEN GO TO STEP2 ;

             PJT EDIT ( 'PAGE NO : ', NPAGE, T1, T2, '     ' )
                  ( PAGE, LINE(2), X(10), A, F(4),
                    2 ( SKIP(1), X(10), A(46) ), SKIP(1), A ) ;
                       GO TO STEP3 ;

STEP2:       PUT EDIT ( 'PAGE NO : ', NPAGE,
                  '*** LISTING OF DESCRIPTOR CATEGORIES AS',
                  '     APPLIED TO VEHICLE DATA ***', '     ' )
                  ( PAGE, LINE(5), X(20), A, F(2),
                    SKIP(3), X(20), A, SKIP(1), X(20), A,
                    SKIP(3), A ) ;

STEP3:       PUT SKIP (1) ;
             END ;

        SIGNAL ENDPAGE (SYSPRINT) ;

   /* HERE, DISPLAY DATASET SETUP, IF REQUIRED */

STEP4:  IF ( DISP = 2 ) THEN GO TO LOOP ;

        ON CONDITION (NEWDP)
             BEGIN ;
             IF ( LT < 61 ) THEN LT = 61 ;

             IF ( LT < 66 ) THEN
LP1:              DO L = LT TO 66 ;
                  DISPLAY ( '     ' ) ;
                  END LP1 ;

             DISPLAY ( '     ' ) ;
             DPAGE = DPAGE + 1 ;
             T3P = DPAGE ;
             DISPLAY ( T3 ) ;
             DISPLAY ( '     ' ) ;
             LT = 4 ;

        IF ( PARM = 'A' ) THEN
TTLA:   DO ;
        DISPLAY ( 'ITEM    STOCK NUM    DESIGNATOR   DEVICE COST' ) ;
        DISPLAY ( '----  -------------  ----------  --------------' ) ;
        DISPLAY ( '     ' ) ;
        LT = LT + 3 ;
        END TTLA ;

             END ;

        SIGNAL CONDITION (NEWDP) ;

/*************************************/

   /* HERE IS MAIN PROGRAM LOOP */

LOOP:   IF ( PARM ¬= 'B' ) THEN GO TO STEP5 ;

   /* HERE, PRINT SETUP FOR B-OPTION - VEHICLE DESIGNATORS */

BLOOP:  NREC = NREC + 1 ;
ER1:    IF ( NREC > MAXC ) THEN GO TO BUST1 ;
        READ FILE (DIND) INTO (DLINE) KEY (NREC) ;
             GJ TO STEP6 ;
 PAGE NUMBER :    4

 SOURCE CARD LISTING FOR
  P8A:     /* ABBREVIATED VEHICLE FILE LISTINGS - WED/2/FEB/77 */


   /* HERE, SETUP FOR A-OPTION, VEHICLE DATA PRINT */

STEP5:  JKEY = JKEY + 1 ;
        IF ( JKEY > MAXSS ) THEN GO TO EOFAB ;
        NREC = VEHPTR (JKEY) ;
ER2:    IF ( NREC > MAXVEH ) THEN GO TO BUST2 ;
        READ FILE (VEHS) INTO (AVEHICLES) KEY (NREC) ;
        NC = NC + 1 ;

/****************************************/

   /* NOW, OUTPUT THE NEW LINE(S); FIRST TRY DISPLAY */

STEP6:  IF ( DISP = 2 ) THEN GO TO STEP7 ;

        IF ( PARM = 'A' ) THEN
LP2:          DO ;
              IF ( LT > MAXLTA ) THEN SIGNAL CONDITION (NEWDP) ;
              DISPLAY ( '    ' ) ;

              PUT STRING (DLINE) EDIT
                    ( NC, STOCKN, DEVDESIG, DEVCOST )
                    ( F(4), X(2), A(13), X(2), A(9),
                      X(2), P'$$$,$$$,$$9.V99' ) ;
              DISPLAY (DLINE) ;

              DISPLAY ( (10)' ' || DEVNAME ) ;

              LT = LT + 3 ;
                    GO TO STEP7 ;
              END LP2 ;

        IF ( LT > MAXLTB ) THEN SIGNAL CONDITION (NEWDP) ;

        IF ( PARM = 'B') & ( (D2 = '**') | (D2 = '--') )
              THEN DO ;  DISPLAY ( '     ' ) ;  LT = LT + 1 ;  END ;
        DISPLAY ( DLINE ) ;

        LT = LT + 1 ;

   /* HERE, TRY FOR PRINTER OUTPUT */

STEP7:  IF ( PTR = 2 ) THEN GO TO TALLY ;

        IF ( PARM = 'B' ) THEN
LP3:          DO ;

              IF ( (D2 = '**') | (D2 = '--') )
                    THEN PUT SKIP (2) ;

              PUT EDIT ( DLINE ) ( SKIP(1), X(10), A(80) ) ;
                    GO TO TALLY ;
              END LP3 ;

        PUT EDIT ( NC, STOCKN, DEVDESIG, DEVCOST, DEVNAME )
              ( SKIP(2), X(10), F(4), X(2), A(13), X(2), A(9),
                X(2), P'$$$,$$$,$$9.V99', X(3), A(47) ) ;

   /* HERE, RECYCLE FOR NEXT, IF REQUIRED */

TALLY:  IF ( PARM = 'A' ) THEN GO TO STEP5 ;

TALLYB: IF ( NREC >= DEND )
              THEN GO TO EOFAB ;
              ELSE GO TO LOOP ;


/****************************************/
```

```
PAGE NUMBER :   5

SOURCE CARD LISTING FOR
 P8A:    /* ABBREVIATED VEHICLE FILE LISTINGS - WED/2/FEB/77 */
```

```


   /* MAX KEYS EXCEEDED ERRORS PROCESSED HERE */

BUST1:  ENO = 1 ;
             GO TO ERCOM ;
BUST2:  ENO = 2 ;

ERCOM:  ENREC = NREC ;

        IF ( DISP = 2 ) THEN
LP4:         DO ;
             DISPLAY ( '     ' ) ;
             DISPLAY ( '     ' ) ;
             DISPLAY ( ELINE ) ;
             DISPLAY ( '     ' ) ;
             END LP4 ;

        IF ( PTR = 1 ) THEN
             PUT LIST ( ELINE ) SKIP (3) ;


/****************************************/


   /* YE OLDE WINDUP HERE */

EOFAB:  CLOSE FILE (MREF), FILE (DIND), FILE (VEHS),
              FILE (DVEH) ;

             END P8A ;
```

```
PAGE NUMBER :   1

SOURCE CARD LISTING FOR
 P9:      /* TASK DATA/DESCR/DIRECTORY LOAD - FRI/28/JAN/77 */

P9:       /* TASK DATA/DESCR/DIRECTORY LOAD - FRI/28/JAN/77 */
          PROC OPTIONS (MAIN) ;

   /*     REVISIONS :
          FRI/28/JAN/77 - INITIAL INSTALLATION.

          WED/02/FEB/77 - CORRECTIONS TO FIX DATA LOAD.

          FRI/18/MAR/77 - MODS PRIOR TO INSTALLATION AT NCSS.
   */

DECLARE CARD              CHAR (80) STATIC,
        CC7079            CHAR (10) DEF CARD POS (70),
        DLINE             CHAR (80) STATIC,
        DL4               CHAR (4) DEF DLINE POS (1),
        ( DISP, PRTR, NPAGE )     FIXED BIN STATIC,
        FTASKS (1200)     FIXED BIN STATIC,
        NODATA (1200)     CHAR (13) STATIC,
        ( ND, NV, NT, K, NSK, TOTD, NS,
              NTEST, TER, NER, NDD, NFIND, NOD, NLIST,
              NDIR, J ) FIXED BIN STATIC ;

DECLARE 1 DINDEX          STATIC,           /* OVERALL DESCRIPTOR INDEX */
          2 DSTART        FIXED BIN,        /* CKF.DESC.INDEX */ .
          2 DEND          FIXED BIN,
          2 NCAT          FIXED BIN,
          2 NDESC         FIXED BIN,
          2 CATEGORIES (15),
            3 PCAT        FIXED BIN,
            3 SCAT        FIXED BIN,
            3 ECAT        FIXED BIN,
          2 PDESC (111) FIXED BIN,
        COVER (4)         CHAR (80) BASED (PCOV) ;

DECLARE 1 MAST            STATIC,
          2 REFS (40)     FIXED BIN,

        DIND              FILE RECORD DIRECT
              KEYED ENV ( REGIONAL(1) ) ;


/****************************************/


DECLARE 1 TDATACARD       STATIC,
          2 F13           CHAR (13),
          2 JTTL          CHAR (50),
          2 FILL          CHAR (6),
          2 BCOST         CHAR (10),
          2 FILL2         CHAR (1),

        TOVER             CHAR (80) BASED (PTO80),
        TCOST             FIXED DEC (10,2) ;

DECLARE 1 TASKCARD        STATIC,           /* TASK DESCRIPTOR CARDS */
          2 TASKN         CHAR (13),        /* CKF.TDESC */
          2 TASKNUMS      CHAR (67),

        1 TASKOVER        BASED (PTO),
          2 RATING        CHAR (5),
          2 RANK          CHAR (2),
          2 JOBTASK       CHAR (6),

        TASK80            CHAR (80) BASED (PTT80),
        TVNS (22)         FIXED BIN STATIC,
        YETASK            CHAR (13) STATIC INIT ( 'ABCDEF' ) ;

DECLARE 1 TASKDESC        STATIC,           /* TASK DESCRIPTOR FILE */
```

```
 PAGE NUMBER :    2

 SOURCE CARD LISTING FOR
  P9:      /* TASK DATA/DESCR/DIRECTORY LOAD - FRI/28/JAN/77 */

             2 TASKDS       CHAR (13),      /* CKF.TASK.DESC */
             2 TTAB (100) FIXED BIN ;

DECLARE 1 TASKDIR (1200) STATIC,          /* JOBTASK FILE DIRECTORY */
             2 CHTASK       CHAR (13),      /* CKF.TASKS.DIR */
             2 TASKPTR      FIXED BIN,

          COVTASK (12)     CHAR (1500) BASED (PTTASK),
          LOADTASK         CHAR (1500) BASED (PLTASK) ;

DECLARE 1 ATASKS           STATIC,          /* ABBR. TASK FILE */
             2 RATING       CHAR (7),       /* CKF.ATASKS */
             2 JOBTASK      CHAR (6),
             2 TASKTTL      CHAR (50),
             2 FILL         CHAR (6),
             2 BILCOST      PICTURE '(8)9V99',
             2 CC80         CHAR (1) ;

/***************************************/

START:  ND, NV, TER, NER, NS,
              TOTD, NFIND, NOD,
              J = 0 ;

          PCOV = ADDR (DINDEX) ;
          PTT80 = ADDR (TASKCARD) ;
          PTTASK = ADDR (TASKDIR) ;
          PTO80 = ADDR (ATASKS) ;
          PTO = ADDR (TDATACARD) ;

          NDIR = -1 ;            /* DIRECTORY RECORD POINTER */
          NOD = 1 ;              /* DIRECTORY LOAD SS, STARTS AT 2 */
          FTASKS = 0 ;           /* TASK FLAG TABLE */
          CHTASK = ' ' ;         /* TASK DIRECTORY PRELIMINARY FILL */
          TASKPTR = 0 ;
          NLIST = 1 ;            /* PRINTER TITLE CONTROL */

          OPEN FILE (MREF) RECORD UPDATE,
               FILE (DIND) INPUT,
               FILE (TDATA) RECORD INPUT,
               FILE (TCDESC) RECORD INPUT,
               FILE (DTASK) RECORD OUTPUT,
               FILE (TDESC) RECORD OUTPUT,
               FILE (TASKS) RECORD OUTPUT ;

          ON ENDFILE (TCDESC) GO TO EOFDESC ;
          ON ENDFILE (TDATA) GO TO EOFDATA ;

          READ FILE (MREF) INTO (MAST) ;

LP1:           DO J = 9 TO 12 ;
               READ FILE (DIND) INTO (CARD) KEY (J) ;
               COVER ( J - 8 ) = CARD ;
               END LP1 ;

          DISPLAY ( '    ' ) ;
          DISPLAY ( 'TASK DATA/DIRECTORY/DESCRIPTOR LOAD'
               || ' PROGRAM IS NOW STARTING' ) ;

/***************************************/

   /* PRINTER/TERMINAL REQUIREMENTS SETUP IN THIS SECTION */

          NPAGE = 0 ;
          DISP, PRTR = 1 ;

          DISPLAY ( '    ' ) ;
TP1:      DISPLAY ( 'OUTPUT TO GO TO TERMINAL, OFFLINE PRINTER,'
               || ' OR BOTH?' ) ;
```

```
PAGE NUMBER :    3

SOURCE CARD LISTING FOR
 P9:       /* TASK DATA/DESCR/DIRECTORY LOAD - FRI/28/JAN/77 */

TP2:     DISPLAY ( 'ENTER TERM, PRTR, BOTH, OR QUIT' ) REPLY (DLINE) ;

         IF ( DL4 = 'TERM' ) THEN GO TO STERM ;
         IF ( DL4 = 'PRTR' ) THEN GO TO OPRTR ;
         IF ( DL4 = 'BOTH' ) THEN GO TO SPRTR ;
         IF ( DL4 = 'QUIT' ) THEN GO TO FEND ;

         DISPLAY ( '     ' ) ;
         DISPLAY ( 'UNABLE TO DETERMINE ENTRY TYPE - TRY AGAIN' ) ;
         DISPLAY ( '     ' ) ;
              GO TO TP2 ;

OPRTR:   DISP = 2 ;

SPRTR:   OPEN FILE (SYSPRINT) LINESIZE (120) PAGESIZE (60) ;

         ON ENDPAGE (SYSPRINT)
              BEGIN ;
              NPAGE = NPAGE + 1 ;
              IF ( NLIST = 2 ) THEN GO TO TTLL2 ;

              PUT EDIT ( 'PAGE NUMBER : ', NPAGE,
                   'NUM    RATE  RANK  TASK   TASK TITLE',
                   '----   -----  --  ------  ----------', '     ' )
                   ( PAGE, X(10), A, F(4),
                     SKIP(2), X(10), A, 2 ( SKIP(1), X(10), A ) ) ;
                        GO TO SKP ;

TTLL2:        PUT EDIT ( 'PAGE NUMBER : ', NPAGE,
                   'NUM    DESCRIPTOR CARD',
                   '----   ---------------', '     ' )
                   ( PAGE, X(10), A, F(4),
                     SKIP(2), X(10), A, 2 ( SKIP(1), X(10), A ) ) ;

SKP:          PUT SKIP (1) ;
              END ;

                   GO TO STARTP ;

STERM:   PRTR = 2 ;

STARTP: IF ( DISP = 2 ) THEN GO TO PROCESS ;

         DISPLAY ( '     ' ) ;
         DISPLAY ( '** LOAD OF TASK ABBREVIATED DATA FILE FOLLOWS' ) ;

         DISPLAY ( '     ' ) ;
         DISPLAY ( 'NUMBER  RATE RANK TASK   TASK TITLE' ) ;
         DISPLAY ( '------  ----- -- ------  ----------' ) ;
         DISPLAY ( '     ' ) ;

/****************************************/

   /* TASK ABBREVIATED FILE DATA LOADED HERE */

PROCESS:
         IF ( PRTR = 1 ) THEN SIGNAL ENDPAGE (SYSPRINT) ;
         IF ( PRTR = 1 ) THEN
         PUT LIST ( '** LOAD OF TASK DATA FILE FOLLOWS **' ) SKIP (2) ;
         IF ( PTR = 1 ) THEN PUT SKIP (3) ;

STEP1:   READ FILE (TDATA) INTO (TDATACARD) ;
         NV = NV + 1 ;

         GET STRING (BCOST)  EDIT (TCOST) ( F(10,2) ) ;
         PUT STRING (BCOST)  EDIT (TCOST) ( P'(8)9V99' ) ;

         IF ( PRTR = 1 ) THEN
         PUT EDIT ( NV, TASKOVER.RATING, TASKOVER.RANK,
 PAGE NUMBER :    4

 SOURCE CARD LISTING FOR
  P9:        /* TASK DATA/DESCR/DIRECTORY LOAD - FRI/26/JAN/77 */

             TASKOVER.JOBTASK, JTTL, BCOST )
             ( SKIP(1), X(10), F(4), X(3), A(5), X(2), A(2), X(2),
               A(6), X(2), A(50), X(3), A(10) ) ;

        IF ( DISP = 2 ) THEN GO TO S1A ;
        PUT STRING (DLINE) EDIT
             ( NV, TASKOVER.RATING, TASKOVER.RANK,
               TASKOVER.JOBTASK, JTTL )
             ( P'ZZ,ZZ9', X(2), A, X(1), A, X(1), A, X(2), A ) ;
        DISPLAY ( DLINE ) ;

S1A:    IF ( F13 = YETASK) THEN
DUPS:        DO ;
             NV = NV - 1 ;

             IF ( PRTR = 1 ) THEN
             PUT EDIT ( '** DUPLICATE RATING/RANK/TASK ON THE ',
                  'PRECEEDING CARD,', '   RECORD WILL BE IGNORED **' )
                  ( SKIP(2), A, A, SKIP(1), A ) ;

             IF ( DISP = 2 ) THEN GO TO STEP1 ;
        DISPLAY ( '    ' ) ;
             DISPLAY ( '** DUPLICATE RATING/RANK/TASK ON THE '
                  || 'PRECEEDING CARD,' ) ;
             DISPLAY ( '   RECORD WILL BE IGNORED' ) ;
        DISPLAY ( '    ' ) ;

             GO TO STEP1 ;
             END DUPS ;

        WRITE FILE (TASKS) FROM (TDATACARD) ;
        YETASK = F13 ;
        NDD = NDD + 1 ;
        NDIR = NDIR + 1 ;
        CHTASK (NDD) = F13 ;
        TASKPTR (NDD) = NDIR ;

                  GO TO STEP1 ;

   /* HERE, EOF ON CARD DATA INPUT DATASET */

EOFDATA: TASKPTR (1) = NDD ;

LP2:         DO J = 1 TO 12 ;
             PLTASK = ADDR (COVTASK(J) ) ;
             WRITE FILE (DTASK) FROM (LOADTASK) ;
             END LP2 ;

        DISPLAY ( '    ' ) ;
        DISPLAY ( '** EOF ON DATA INPUT FILE, ' ) ;
        PUT STRING (DLINE) EDIT
             ( '   CARDS PROCESSED : ', NV, ' **' )
             ( A, P'ZZ,ZZ9', A ) ;
        DISPLAY ( DLINE ) ;

        DISPLAY ( '    ' ) ;
        DISPLAY ( '** LOAD OF TASK DESCRIPTOR CARDS FOLLOWS **' ) ;
        DISPLAY ( '    ' ) ;

        REFS (11) = NV - 1 ;
        REFS (12) = NDD ;

/****************************************/

   /* INITILIZE START OF DESCRIPTOR LOAD HERE */

        IF ( PRTR = 2 ) THEN GO TO PHAZE2 ;

        NLIST = 2 ;
 PAGE NUMBER :    5

 SOURCE CARD LISTING FOR
  P9:     /* TASK DATA/DESCR/DIRECTORY LOAD - FRI/28/JAN/77 */

             SIGNAL ENDPAGE (SYSPRINT) ;
        PUT LIST ( '** LOAD OF DESCRIPTOR CARD DATA FOLLOWS **' )
             SKIP (2) ;
        PUT SKIP (3) ;

PHAZE2: READ FILE (TCDESC) INTO (TASKCARD) ;
        ND = 1 ;
             GO TO STEP3 ;

   /* HERE IS THE DESCRIPTOR LOAD LOOP */

STEP2:  READ FILE (TCDESC) INTO (TASKCARD) ;
        ND = ND + 1 ;

        IF ( TASKN = YETASK ) THEN
LP3:         DO ;
             NSK = 1 ;
                  GO TO STEP4 ;
             END LP3 ;

        WRITE FILE (TDESC) FROM (TASKDESC) ;
        NS = NS + 1 ;

STEP3:  YETASK = TASKN ;
        TOTD = TOTD + 1 ;
        TTAB = 0 ;
        TASKDS = TASKN ;
             NSK = 2 ;

LP5:         DO J = 1 TO NOD ;    /* SEE IF MATCHING DATA AVAILABLE */
             IF ( TASKN ¬= CHTASK(J) ) THEN GO TO LP5END ;
             NFIND = NFIND + 1 ;
             FTASKS (J) = 1 ;
                       GO TO STEP4 ;
LP5END:      END LP5 ;

        NOD = NOD + 1 ;
        NODATA (NOD) = TASKN ;

STEP4:  IF ( PRTR = 1 ) THEN
        PUT EDIT ( ND, TASKCARD )
             ( SKIP(1),   X(10), F(4), X(3), A(13), A(67) ) ;

        IF ( DISP = 1 ) THEN DISPLAY ( TASK80 ) ;

        TVNS = 0 ;
        GET STRING (TASKNUMS) EDIT (TVNS) ( 22 ( F(2), X(1) ) ) ;

             NT = 0 ;
LP4:         DO J = 1 TO 22 ;
             NTEST = TVNS (J) ;
             IF ( NTEST = 0 ) THEN GO TO ENDLP4 ;

                  NT = NT + 1 ;
             IF ( PDESC(NTEST) ¬= 0 ) THEN GO TO STEP5 ;

        NER = NER + 1 ;

        IF ( PRTR = 1 ) THEN
        PUT EDIT ( '** ERROR IN TASK NUMBER -', TASKN,
             '- DESCRIPTOR SUBSCRIPT ', NTEST,
             ' DOES NOT EXIST **' )
             ( SKIP(1), A, A(13), A, F(3), A ) ;

        IF ( DISP = 2 ) THEN GO TO BUMD ;
        PUT STRING (DLINE) EDIT
                  ( '** ERROR IN TASK NUMBER -', TASKN,
             '- DESCRIPTOR SUBSCRIPT ', NTEST,
             ' DOES NOT EXIST **' )
 PAGE NUMBER :   6

 SOURCE CARD LISTING FOR
  P9:     /* TASK DATA/DESCR/DIRECTORY LOAD - FRI/28/JAN/77 */

              ( A, A, A, F(3), A ) ;
         DISPLAY ( '     ' ) ;   DISPLAY ( DLINE ) ;
         DISPLAY ( '     ' ) ;

BUMD:    IF ( NER > 100 ) THEN GO TO EOJ ;
              GO TO ENDLP4 ;

STEP5:        TTAB (NTEST) = PDESC (NTEST) ;
ENDLP4:       END LP4 ;

                   GO TO STEP2 ;

   /* HERE IS EOF ON DESCRIPTOR CARD INPUT */

EOFDESC: WRITE FILE (TDESC) FROM (TASKDESC) ;
          NS = NS + 1 ;

         REFS (13) = NS ;
         REFS (14) = ND ;

         DISPLAY ( '     ' ) ;
         DISPLAY ( '** EOF REACHED ON TASK DESCRIPTOR INPUT FILE,' ) ;
         PUT STRING (DLINE) EDIT
              ( '   CARDS PROCESSED : ', ND, ' **' )
              ( A, P'ZZ,ZZ9', A ) ;
         DISPLAY ( DLINE ) ;

/****************************************/

   /* ANY REQUIRED DIAGNOSTICS OUTPUT IN THIS SECTION */

          IF ( NFIND = NV ) THEN GO TO STEP6 ;

   /* HERE DESCRIPTOR / DATA MISMATCHES PRINTED */

         IF ( PRTR = 2 ) THEN GO TO LP6A ;
         PUT EDIT ( '** THE FOLLOWING DATA FILE TASK NUMBERS',
              '    HAVE NO MATCH IN THE DESCRIPTOR CARD FILE **' )
              ( SKIP(3), A, SKIP(1), A ) ;
         PUT SKIP (2) ;

LP6A:    DISPLAY ( '     ' ) ;            /* NOTE - ALWAYS DISPLAY */
         DISPLAY ( '** THE FOLLOWING DATA FILE TASK NUMBERS' ) ;
         DISPLAY ( '    HAVE NO MATCH IN THE DESCRIPTOR CARD FILE **');
         DISPLAY ( '     ' ) ;

              K = 0 ;
LP6:          DO J = 1 TO NDD ;
              IF ( FTASKS(J) = 1 ) THEN GO TO LP6END ;
              K = K + 1 ;

              PUT STRING (DLINE) EDIT ( K, CHTASK(J) )
              ( F(3), X(2), A ) ;
              DISPLAY ( DLINE ) ;

              IF ( PRTR = 1 ) THEN
                   PUT EDIT ( K, CHTASK(J) )
                         ( SKIP(1), X(10), F(3), X(3), A ) ;
LP6END:       END LP6 ;

   /* HERE, DESCRIPTOR TASK NUMBERS W/O MATCHING DATA FILE ENTRIES
      ARE PRINTED - IF ANY */

STEP6:   IF ( NDD = 0 ) THEN GO TO BITTEREND ;

         IF ( PRTR = 2 ) THEN GO TO LP7A ;
         PUT EDIT ( '** THE FOLLOWING DESCRIPTOR CARD TASK NUMBERS ',
              '     HAVE NO MATCH IN THE DATA FILE **' )
              ( SKIP(3), A, SKIP(1), A ) ;
```

```
PAGE NUMBER :    7

SOURCE CARD LISTING FOR
 P9:     /* TASK DATA/DESCR/DIRECTORY LOAD - FRI/28/JAN/77 */

      PUT SKIP(2) ;

LP7A: DISPLAY ( '    ' ) ;   DISPLAY ( '    ' ) ;
      DISPLAY ( '** THE FOLLOWING DESCRIPTOR CARD TASK NUMBERS ' ) ;
      DISPLAY ( '   HAVE NO MATCH IN THE DATA FILE **' ) ;
      DISPLAY ( '    ' ) ;

LP7:          DO J = 1 TO NOD ;

              IF ( PRTR = 1 ) THEN
              PUT EDIT ( J, NODATA(J) ) ( SKIP(1), X(10), F(3),
                    X(3), A(13) ) ;

              PUT STRING (DLINE) EDIT ( J, NODATA(J) )
                    ( F(3), X(2), A ) ;
              DISPLAY ( DLINE ) ;

              END LP7 ;

/****************************************/

  /* HERE AM THE WINDUP */

BITTEREND:
      REWRITE FILE (MREF) FROM (MAST) ;

FEND:
EOJ:  CLOSE FILE (MREF), FILE (DIND), FILE (TDATA),
              FILE (TCDESC), FILE (DTASK), FILE (TDESC),
              FILE (TASKS) ;

      DISPLAY ( '    ' ) ;
      DISPLAY ( 'TASK FILE DATA/DIRECTORY/DESCRIPTOR LOAD PROGRAM'
              || ' IS NOW TERMINATING' ) ;

              END P9 ;
 PAGE NUMBER :   1

 SOURCE CARD LISTING FOR
  P9A:     /* ABBREVIATED TASK FILE LISTINGS - FRI/4/FEB/77 */


P9A:    /* ABBREVIATED TASK FILE LISTINGS - FRI/4/FEB/77 */
        PROC (INPARM) OPTIONS (MAIN) ;

   /*   REVISIONS :
        FRI/04/FEB/77 - FIRST IMPLEMENTATION.

        SUN/20/MAR/77 - MODS PRIOR TO INSTALLATION AT NCSS.
   */

DECLARE ( DISP, PTR, NPAGE, DPAGE, LT, JKEY, LP, L, MAXC,
             MAXIND, MAXLT, MAXLTA, MAXLTB, MAXSS,
             MAXTASK, NC, NREC,
             J, K )    FIXED BIN STATIC ;

DECLARE ( DIND, TASKS )     FILE RECORD DIRECT
                  KEYED ENV ( REGIONAL(1) ),

        INPARM         CHAR (100) VARYING,
        1 POVER        BASED (PP),
          2 PARM       CHAR (1),
          2 PARM2      CHAR (4),

        TCOST          FIXED DEC (10,2) STATIC,
        ( DLINE, CARD ) CHAR (80) STATIC,
        D2             CHAR (2) DEF DLINE POS (1),
        BLANK          CHAR (4) STATIC INIT ( '    ' ),

        T1             CHAR (45) STATIC INIT (
'ITEM    RATING  RANK  JOBTASK    BILLET COST' ),

        T2             CHAR (45) STATIC INIT (
'-----   ------  ----  -------   -------------' ),

        ELINE          CHAR (39) STATIC INIT (
'** ERROR XX OCCURED ; NREC WAS XXXXX **' ),
        ENO            CHAR (2) DEF ELINE POS (10),
        ENREC          PICTURE 'Z,ZZ9' DEF ELINE POS (32),

        T3             CHAR (11) STATIC INIT (
'PAGE : XXXX' ),
        T3P            PICTURE 'ZZZ9' DEF T3 POS (8) ;

/****************************************/

DECLARE 1 DINDEX       STATIC,        /* OVERALL DESCRIPTOR INDEX */
          2 DSTART     FIXED BIN,     /* CKF.DESC.INDEX */
          2 DEND       FIXED BIN,
          2 NCAT       FIXED BIN,
          2 NDESC      FIXED BIN,
          2 CATEGORIES (15),
            3 PCAT     FIXED BIN,
            3 SCAT     FIXED BIN,
            3 ECAT     FIXED BIN,
          2 PDESC (111) FIXED BIN,
        COVER (4)      CHAR (80) BASED (PCOV) ;

DECLARE 1 MAST         STATIC,
          2 REFS (40)  FIXED BIN ;

DECLARE 1 ATASKS       STATIC,        /* ABBR. TASK FILE */
          2 RATING     CHAR (7),      /* CKF. ATASKS */
          2 JOBTASK    CHAR (6),
          2 TASKTTL    CHAR (50),
          2 FILL       CHAR (6),
          2 BILCOST    PICTURE '(8)9V99',
          2 CC80       CHAR (1) ;

DECLARE 1 TASKOVER     BASED (PTO),
 PAGE NUMBER :    2

 SOURCE CARD LISTING FOR
  P9A:    /* ABBREVIATED TASK FILE LISTINGS - FRI/4/FEB/77 */

          2 RATING     CHAR (5),
          2 RANK       CHAR (2),
          2 JOBTASK    CHAR (6) ;

DECLARE 1 TASKDIR (1200) STATIC,      /* JOBTASK FILE DIRECTORY */
          2 CHTASK     CHAR (13),     /* CKF.TASKS.DIR */
          2 TASKPTR    FIXED BIN,

        COVTASK (12)   CHAR (1500) BASED (PTTASK),
        LOADTASK       CHAR (1500) BASED (PLTASK) ;

/****************************************/

START:  PP = ADDR (INPARM) ;
        PTO = ADDR (ATASKS) ;
        PCOV = ADDR (DINDEX) ;
        PTTASK = ADDR (TASKDIR) ;

        NPAGE, DPAGE, NC,
             J, K = 0 ;

        LT = 100 ;          /* DISPLAY PAGE LINE COUNTER */
        MAXLT = 60 ;        /* MAX # LINES ON DISPLAY */
        JKEY = 1 ;          /* DIRECTORY SS, SKIPS DIR(1) */

COMMENCE:  OPEN FILE (MREF) RECORD INPUT,
             FILE (DIND) INPUT,
             FILE (TASKS) INPUT,
             FILE (DTASK) RECORD INPUT ;

        READ FILE (MREF) INTO (MAST) ;
        MAXTASK = REFS (11) ;
        MAXC = REFS (4) ;
        MAXSS = REFS (12) ;

        IF ( PARM = 'B' ) THEN GO TO BSETUP ;

   /* A-OPTION, PRINT DATA FILE SETUP HERE */

        MAXLTA = MAXLT - 3 ;

DIRLOAD:     DO J = 1 TO 12 ;
             PLTASK = ADDR ( COVTASK(J) ) ;
             READ FILE (DTASK) INTO (LOADTASK) ;
             END DIRLOAD ;

                  GO TO FIRST ;

   /* B-OPTION ROUTE INITILIZED HERE */

BSETUP:      DO J = 9 TO 12 ;
             READ FILE (DIND) INTO (CARD) KEY (J) ;
             COVER ( J - 8 ) = CARD ;
             END BSETUP ;

        MAXIND = REFS (4) ;
        NREC = DSTART - 1 ;
        MAXLTB = MAXLT - 1 ;

/****************************************/

   /* NOW INITILIZE DEVICE ASSIGNMENTS */

FIRST:  DISP, PTR = 1 ;

        IF ( PARM2 = 'BOTH' ) THEN GO TO STEP1 ;
        IF ( PARM2 = 'TERM' )
             THEN PTR = 2 ;
             ELSE DISP = 2 ;
 PAGE NUMBER :   3

 SOURCE CARD LISTING FOR
  P9A:    /* ABBREVIATED TASK FILE LISTINGS - FRI/4/FEB/77 */

STEP1:  LT, LP = 100 ;

   /* HERE, PRINTER DATASET IS INITILILED */

        IF ( PTR = 2 ) THEN GO TO STEP4 ;

        OPEN FILE (SYSPRINT)
             LINESIZE (120) PAGESIZE (58) ;

        ON ENDPAGE (SYSPRINT)
             BEGIN ;
             NPAGE = NPAGE + 1 ;

             IF ( PARM = 'B' ) THEN GO TO STEP2 ;

             PUT EDIT ( 'PAGE NO : ', NPAGE, T1, T2, '     ' )
                  ( PAGE, LINE(2), X(10), A, F(4),
                    2 ( SKIP(1), X(10), A(45) ), SKIP(1), A ) ;
                       GO TO STEP3 ;

STEP2:       PUT EDIT ( 'PAGE NO : ', NPAGE,
                  '*** LISTING OF DESCRIPTOR CATEGORIES AS',
                  '     APPLIED TO JOBTASK DATA ***', '     ' )
                  ( PAGE, LINE(5), X(20), A, F(2),
                    SKIP(3), X(20), A, SKIP(1), X(20), A,
                    SKIP(3), A ) ;

STEP3:       PUT SKIP (1) ;
             END ;

        SIGNAL ENDPAGE (SYSPRINT) ;

   /* HERE, SETUP DISPLAY DATASET, IF REQUIRED */

STEP4:  IF ( DISP = 2 ) THEN GO TO LOOP ;

        ON CONDITION (NEWDP)
             BEGIN ;

             IF ( LT < 66 ) THEN
LP1:              DO L = LT TO 66 ;
                  DISPLAY ( BLANK ) ;
                  END LP1 ;

             DISPLAY (BLANK) ;
             DPAGE = DPAGE + 1 ;
             T3P = DPAGE ;
             DISPLAY (T3) ;
             DISPLAY (BLANK) ;
             LT = 4 ;

        IF ( PARM = 'A' ) THEN
ATTL:   DO ;
        DISPLAY ( 'ITEM   RATE  RANK  TASK   COST-PER-BILLET' ) ;
        DISPLAY ( '----   ----- ---- ------  ---------------' ) ;
        DISPLAY ( '     ' ) ;
        LT = LT + 3 ;
        END ATTL ;

             END ;

        SIGNAL CONDITION (NEWDP) ;

/****************************************/

   /* MAINLINE LOOP FOLLOWS */

LOOP:   IF ( PARM ¬= 'B' ) THEN GO TO STEP5 ;
```

```
 PAGE NUMBER :    4

 SOURCE CARD LISTING FOR
  P9A:    /* ABBREVIATED TASK FILE LISTINGS - FRI/4/FEB/77 */

   /* HERE IS B-SECTION PRINT SETUP */

BLOOP:  NREC = NREC + 1 ;
ER1:    IF ( NREC > MAXC ) THEN GO TO BUST1 ;
        READ FILE (DIND) INTO (DLINE) KEY (NREC) ;
             GO TO STEP6 ;

   /* NOW, FOR A-OPTION SETUP */

STEP5:  JKEY = JKEY + 1 ;

        IF ( JKEY > MAXSS ) THEN GO TO EOFAB ;
        NREC = TASKPTR (JKEY) ;

ER2:    IF ( NREC > MAXTASK ) THEN GO TO BUST2 ;
        READ FILE (TASKS) INTO (ATASKS) KEY (NREC) ;
        NC = NC + 1 ;

   /* NOW, FOR NEW LINE(S) OUTPUT ; FIRST TRY TERMINAL */

STEP6:  IF ( DISP = 2 ) THEN GO TO STEP7 ;

        IF ( PARM = 'A' ) THEN
LP2:         DO ;
             IF ( LT > MAXLTA ) THEN SIGNAL CONDITION (NEWDP) ;
             DISPLAY ( BLANK ) ;

             PUT STRING (DLINE) EDIT
                  ( NC, TASKOVER.RATING, TASKOVER.RANK,
                    TASKOVER.JOBTASK, BILCOST )
                  ( F(4), X(3), A(5), X(2), A(2), X(2),
                    A(6), X(2), P'$$$,$$$,$$9.V99' ) ;
             DISPLAY (DLINE) ;

             DISPLAY ( (10)' ' || TASKTTL ) ;

             LT = LT + 3 ;
                  GO TO STEP7 ;
             END LP2 ;

        IF ( LT > MAXLTB ) THEN SIGNAL CONDITION (NEWDP) ;

        IF ( (D2 = '**') | (D2 = '--') ) THEN
             DO ;  DISPLAY ( '    ' ) ;  LT = LT + 1 ;  END ;

        DISPLAY ( DLINE ) ;
        LT = LT + 1 ;

   /* HERE, TRY FOR PRINTER OUTPUT */

STEP7:  IF ( PTR = 2 ) THEN GO TO TALLY ;

        IF ( PARM = 'B' ) THEN
LP3:         DO ;

             IF ( (D2 = '**') | (D2 ='--') )
                  THEN PUT SKIP (1) ;

             PUT EDIT ( DLINE ) ( SKIP(1), X(10), A(80) ) ;
                  GO TO TALLY ;
             END LP3 ;

        PUT EDIT ( NC, TASKOVER.RATING, TASKOVER.RANK,
             TASKOVER.JOBTASK, BILCOST, TASKTTL )
             ( SKIP(2), X(10), P'Z,ZZ9', X(3), A(5), X(4), A(2),
               X(3), A(6), X(3), P'$$$,$$$,$$9.V99', X(3), A(50) ) ;

   /* HERE, RECYCLE FOR NEXT RECORD, IF ANY */
```

```
PAGE NUMBER :   5

SOURCE CARD LISTING FOR
 P9A:    /* ABBREVIATED TASK FILE LISTINGS - FRI/4/FEB/77 */

TALLY:  IF ( PARM = 'A' ) THEN GO TO STEP5 ;

TALLYB: IF ( NREC >= DEND ) THEN GO TO EOFAB ;
                            ELSE GO TO LOOP ;

/****************************************/

   /* MAX KEY EXCEEDED ERRORS PROCESSED IN THIS SECTION */

BUST1:  ENO = '01' ;
              GO TO ERCOM ;
BUST2:  ENO = '02' ;
ERCOM:  ENREC = NREC ;

        IF ( DISP = 2 ) THEN
LP4:          DO ;
              DISPLAY (BLANK) ;
              DISPLAY (BLANK) ;
              DISPLAY (ELINE) ;
              DISPLAY (BLANK) ;
              END LP4 ;

        IF ( PTR = 1 ) THEN
              PUT LIST ( ELINE ) SKIP (3) ;

/****************************************/

   /* HERE IS PROGRAM WINDUP */

EOFAB:  CLOSE FILE (DIND), FILE (MREF), FILE (TASKS),
               FILE (DTASK) ;

              END P9A ;
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
 P14:     /* NITRAS CDP EXTRACT VIA DESCRIPTOR CIN - MON/13/DEC/76 */


P14:      /* NITRAS CDP EXTRACT VIA DESCRIPTOR CIN - MON/13/DEC/76 */
          PROC OPTIONS (MAIN) ;

   /*     REVISIONS :
          TUE/11/JAN/77 - RAN VERYFIRST TIME .

          MON/14/MAR/77 - MODS TO PREPARE FOR SHIP TO NCSS.
   */

DECLARE 1 RNIT,          /* ORIGINAL NITRAS TAPE RECORD */
            2 F1    CHAR (36),
            2 FILL1 CHAR (6),
            2 PC    CHAR (2),
            2 FILL2 CHAR (1),
            2 F2    CHAR (7),
            2 FILL3 CHAR (19),
            2 STCD  CHAR (1),
            2 STDTE FIXED DEC (5),
            2 FILL4 CHAR (8),
            2 ATTR  FIXED DEC (3,1),
            2 STBK  FIXED DEC (3,1),
            2 THRS  FIXED DEC (5),
            2 LHRS  FIXED DEC (5),
            2 FILL5 CHAR (3),
            2 F3    CHAR (6),
            2 FILL6 CHAR (42),
            2 CLEN  FIXED DEC (5),
            2 FILL7 CHAR (663) ;

 DECLARE 1 RACRS,                           /* ABBREVIATED COURSE RECORD */
            2 F1    CHAR (36),              /*    (CARD-IMAGE)           */
            2 PC    CHAR (2),
            2 F2    CHAR (7),
            2 STCD  CHAR (1),
            2 STDTE PICTURE '(5)9',
            2 F3    CHAR (6),
            2 ATTR  PICTURE '99V9',
            2 STBK  PICTURE '99V9',
            2 CLEN  PICTURE '999',
            2 THRS  PICTURE '999',
            2 LHRS  PICTURE '999',
            2 CAOB  PICTURE '(5)9V99',
            2 CC80  CHAR (1) ;

 DECLARE 1 OVERLAY        BASED (POV),
            2 TAPCDP      CHAR (4),
            2 TAPCIN      CHAR (8),
            2 TAPCST      CHAR (16) ;

 DECLARE ( TC, CC, NTOT, NDONE, DDONE, NSKP, TOTCIN, CINER,
           NC )    FIXED BIN STATIC,
         ( CCIN, TCIN, PFIRST )   CHAR (8) STATIC,
         ( DLINE, CARD )      CHAR (80) STATIC,

         DCARD      CHAR (80) STATIC,
         CRDCIN     CHAR (8) DEF DCARD,

         SRET       LABEL ;

 DECLARE T1         CHAR (44) STATIC INIT (
    'DESCRIPTOR   CIN               NITRAS   COURSE' ),

         T2         CHAR (64) STATIC INIT (
   ' CARD NUM   COUNT     CIN    REC NUM    CDP   COURSE SHORT TITLE');

 DECLARE RDATE      CHAR (6) STATIC,
         GDATE      CHAR (8) STATIC,
         RTIME      CHAR (9) STATIC,
 PAGE NUMBER :    2

 SOURCE CARD LISTING FOR
  P14:     /* NITRAS CDP EXTRACT VIA DESCRIPTOR CIN - MON/13/DEC/76 */
          GTIME      CHAR (4) STATIC ;

FMT1:     FORMAT ( X(9), P'Z,ZZ9', A ) ;

/****************************************/

START:    CC80 = ' ' ;
          CAOB = 0 ;
          POV = ADDR (RNIT) ;
          NC, NDONE, DDONE, TOTCIN, CINER = 0 ;

          RDATE = DATE ;
          GDATE = SUBSTR (RDATE,3,2) || '/' ||
                  SUBSTR (RDATE,5,2) || '/' ||
                  SUBSTR (RDATE,1,2) ;

          RTIME = TIME ;
          GTIME = SUBSTR (RTIME,1,4) ;

          OPEN FILE (NITRAS) RECORD INPUT,
               FILE (DESCR) RECORD INPUT,
               FILE (COURSE) RECORD OUTPUT ;

          ON ENDFILE (NITRAS) GO TO EOFN ;
          ON ENDFILE (DESCR) GO TO EOFD ;

/****************************************/

          DISPLAY ( '     ' ) ;
          DISPLAY ( 'ABBREVIATED COURSE DATA BASE EXTRACT PROGRAM' ) ;
          DISPLAY ( '        DATE : ' || GDATE ) ;
          DISPLAY ( '        TIME : ' || GTIME ) ;

          DISPLAY ( '     ' ) ;
          DISPLAY ( T1 ) ;    DISPLAY ( T2 ) ;
          DISPLAY ( '     ' ) ;

/****************************************/

          READ FILE (NITRAS) INTO (RNIT) ;
               TCIN = TAPCIN ;
               TC, CC = 1 ;

          READ FILE (DESCR) INTO (DCARD) ;
STEP1:         CCIN, PFIRST = CRDCIN ;

TEST1:    IF ( TCIN ¬= CCIN ) THEN GO TO TEST2 ;

YES:           RACRS = RNIT, BY NAME ;
               WRITE FILE (COURSE) FROM (RACRS) ;
               NC = NC + 1 ;

          IF ( PFIRST = (8)' ' ) THEN GO TO STEP2 ;

               TOTCIN = TOTCIN + 1 ;

          PUT STRING (DLINE) EDIT
               ( CC, TOTCIN, PFIRST, TC, TAPCDP, TAPCST )
               ( X(11), P'Z,ZZ9', X(5), P'Z,ZZ9', X(2), A(8),
                 X(3), P'Z,ZZ9', X(4), A(4), X(4), A(16) ) ;
          DISPLAY ( '     ' ) ;    DISPLAY ( DLINE ) ;

               PFIRST = ' ' ;
                    GO TO STEP3 ;

STEP2:    PUT STRING (DLINE) EDIT
               ( TC, TAPCDP, TAPCST )
               ( X(39), P'Z,ZZ9', X(4), A(4), X(4), A(16) ) ;
          DISPLAY ( DLINE ) ;
 PAGE NUMBER :    3

 SOURCE CARD LISTING FOR
  P14:     /* NITRAS CDP EXTRACT VIA DESCRIPTOR CIN - MON/13/DEC/76 */


STEP3:  READ FILE (NITRAS) INTO (RNIT) ;
                TC = TC + 1 ;

        IF ( TAPCIN = TCIN ) THEN GO TO YES ;
                TCIN = TAPCIN ;

   /* HERE, TAPE CIN HAS ADVANCED */

STEP4:  READ FILE (DESCR) INTO (DCARD) ;
                CC = CC + 1 ;

        IF ( CRDCIN = CCIN ) THEN GO TO STEP4 ;
                GO TO STEP1 ;

   /* HERE, CINS NO COMPARE; TRY CCIN BEHIND TCIN */

TEST2:  IF ( CCIN > TCIN ) THEN GO TO STEP5 ;

                SRET = STEP4 ;
                     GO TO CINGONE ;

   /* HERE, TAPE CIN IS BEHIND CARD CIN */

STEP5:  READ FILE (NITRAS) INTO (RNIT) ;
                TC = TC + 1 ;
                TCIN = TAPCIN ;
                     GO TO TEST1 ;

/****************************************/

   /* HERE, EOF ON NITRAS TAPE DATASET */

EOFN:   NDONE = TC ;
        SRET = STEP6 ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( '** EOF ON NITRAS-TAPE INPUT **' ) ;

        IF ( DDONE > 0 ) THEN GO TO QUIT ;

        IF ( PFIRST = (8)' ' ) THEN GO TO CINGONE ;

STEP6:  READ FILE (DESCR) INTO (DCARD) ;
                CC = CC + 1 ;

        /* NOTE - FALLTHRU TO CINGONE SUBROUTINE, BELOW */

/****************************************/

   /* THIS IS THE MISSING CIN MSG SUBROUTINE ; RETURN = SRET */

CINGONE:
                TOTCIN = TOTCIN + 1 ;

        PUT STRING (DLINE) EDIT
                ( '****', CC, TOTCIN, CCIN, '- NO CIN MATCH IN NITRAS -' )
                ( X(4), A, X(3), P'Z,ZZ9', X(5), P'Z,ZZ9',
                  X(2), A(8), X(2), A ) ;
        DISPLAY ( '     ' ) ;   DISPLAY ( DLINE ) ;

                CINER = CINER + 1 ;

                GO TO SRET ;

/****************************************/

   /* HERE, EOF ON CARD-DESCRIPTOR INPUT */

 PAGE NUMBER :    4

 SOURCE CARD LISTING FOR
  P14:     /* NITRAS CDP EXTRACT VIA DESCRIPTOR CIN - MON/13/DEC/76 */

EOFD:    DDONE = CC ;

         DISPLAY ( '     ' ) ;
         DISPLAY ( '** EOF ON CARD-DESCRIPTOR INPUT **' ) ;

               GO TO QUIT ;

/*****************************************/

QUIT:    DISPLAY ( '     ' ) ;   DISPLAY ( '     ' ) ;

         PUT STRING (DLINE) EDIT
               ( CC, ' : TOTAL DESCRIPTOR CARDS READ' ) (R(FMT1)) ;
         DISPLAY ( DLINE ) ;

         PUT STRING (DLINE) EDIT
               ( TOTCIN, ' : TOTAL CIN NUMBERS INPUT' ) (R(FMT1)) ;
         DISPLAY ( DLINE ) ;

         PUT STRING (DLINE) EDIT
               ( CINER, ' : TOTAL CIN ERRORS (NO MATCH IN NITRAS)' )
                     ( R(FMT1) ) ;
         DISPLAY ( DLINE ) ;

         PUT STRING (DLINE) EDIT
               ( NC, ' : TOTAL COURSES EXTRACTED' ) (R(FMT1)) ;
         DISPLAY ( DLINE ) ;

         PUT STRING (DLINE) EDIT
               ( TC, ' : LAST RECORD READ IN NITRAS' ) (R(FMT1)) ;
         DISPLAY ( DLINE ) ;
         DISPLAY ( '     ' ) ;

         CLOSE FILE (NITRAS), FILE (DESCR), FILE (COURSE) ;

         DISPLAY ( '     ' ) ;
         DISPLAY ( 'COURSE DATABASE CONSTRUCT PROGRAM '
               || 'IS NOW TERMINATING' ) ;
         DISPLAY ( '     ' ) ;

               END P14 ;
```

```
PAGE NUMBER :    1

SOURCE CARD LISTING FOR
 P16:    /* DESCRIPTOR MASTER INDEX MAKEUP - FRI/14/JAN/77 */

P16:    /* DESCRIPTOR MASTER INDEX MAKEUP - FRI/14/JAN/77 */
        PROC OPTIONS (MAIN) ;

   /*   REVISIONS :
        FRI/14/JAN/77 - FIRST VERSION IMPLEMENTED.

        MON/14/MAR/77 - MODS TO PREPARE FOR SHIP TO NCSS.
   */

DECLARE 1 DINDEX        BASED (PIN),
          2 DSTART      FIXED BIN,
          2 DEND        FIXED BIN,
          2 NCAT        FIXED BIN,
          2 NDESC       FIXED BIN,
          2 CATEGORIES (15),
             3 PCAT     FIXED BIN,
             3 SCAT     FIXED BIN,
             3 ECAT     FIXED BIN,
          2 PDESC (111) FIXED BIN,

        ( ICOURSE(160), IVEH(160), ITASK(160) )
             FIXED BIN STATIC,
        COVER (4)       CHAR (80) BASED (PCOV) ;

DECLARE CARD            CHAR (80) STATIC,
             CF2        CHAR (2) DEF CARD POS (1),
             CTYP       CHAR (8) DEF CARD POS (4),
             TREC       PICTURE '9999' DEF CARD,
        FLAG (3)        BIT (1) STATIC,
        ( NREC, NER, NOUT, NCC, NC, NPAGE,
             J, K, L ) FIXED BIN STATIC ;

DECLARE 1 MAST          STATIC,
          2 REFS (40)   FIXED BIN ;

DECLARE TTESTS (3)      CHAR (8) STATIC INIT
             ( 'COURSES', 'VEHICLES', 'TASKS' ),
        BLANK           CHAR (4) STATIC INIT ( '    ' ),
        DLINE           CHAR (80) STATIC,
        POINT (3)       POINTER,
        ( ROUTE, RPRINT ) LABEL ;

DECLARE 1 CARBIN        STATIC,
          2 CNUMS (40)  FIXED BIN ;

DECLARE EMSGS (7)       CHAR (38) STATIC INIT (
            'FIRST CARD NOT (--) TYPE CARD',
            'DUPLICATE (--) CARD',
            'TYPE OF (--) CARD INVALID',
            'TYPE (**) CARD MUST FOLLOW TYPE (--)',
            'CATEGORY HAS NO DESCRIPTORS',
            'DESCRIPTOR NUMBER CONVERSION ERROR',
            'CATEGORY TABLE SIZE EXCEEDED, MAX = 15' ) ;

/*************************************/

START:  ON ERROR GO TO QUIT ;

        OPEN FILE (CARDS) RECORD INPUT ;
        ON ENDFILE (CARDS) GO TO EOFCARDS ;

        NREC = 13 ;
        NER, NC,
             NPAGE = 0 ;
        ICOURSE, IVEH, ITASK = 0 ;
        FLAG = '0'B ;

        POINT (1) = ADDR (ICOURSE) ;
```

```
PAGE NUMBER :    2

SOURCE CARD LISTING FOR
 P16:     /* DESCRIPTOR MASTER INDEX MAKEUP - FRI/14/JAN/77 */

        POINT (2) = ADDR (IVEH) ;
        POINT (3) = ADDR (ITASK) ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( 'MASTER DESCRIPTOR INDEX CONSTRUCT '
             || 'PROGRAM IS NOW STARTING' ) ;

/*************************************/

   /* VERY FIRST CARD PROCESSED HERE */

        READ FILE (CARDS) INTO (CARD) ;
ER1:    NER, NC = 1 ;
        IF ( CF2 ¬= '--' ) THEN GO TO QUIT ;

   /* HERE, CONTENTS OF (--) CARD VERIFIED */

STEP1:       DO K = 1 TO 3 ;
             IF ( CTYP ¬= TTESTS(K) )
                  THEN GO TO ENDS1 ;

             J = K ;
ER2:         NER = 2 ;
             IF ( FLAG(J) ) THEN GO TO QUIT ;

                  PIN = POINT(J) ;
                  FLAG(J) = '1'B ;
                       GO TO PASS1 ;
ENDS1:       END STEP1 ;

ER3:    NER = 3 ;
             GO TO QUIT ;

   /* HERE, START PROCESSING NEW (--) TYPE */

PASS1:  DSTART = NREC ;
R1:     READ FILE (CARDS) INTO (CARD) ;
        NREC = NREC + 1 ;  NC = NC + 1 ;
        IF ( CF2 = '  ' ) THEN GO TO R1 ;

ER4:    NER = 4 ;
        IF ( CF2 ¬= '**' ) THEN GO TO QUIT ;

   /* HERE, STARTING TO PROCESS NEW (**) CATEGORY */

STEP2:  NCAT = NCAT + 1 ;
             PCAT (NCAT) = NREC ;

R2:     READ FILE (CARDS) INTO (CARD) ;
             NREC = NREC + 1 ;  NC = NC + 1 ;
             IF ( CF2 = '  ' ) THEN GO TO R2 ;

ER5:    NER = 5 ;
        IF ( CF2 = '**' ) | ( CF2 = '--' ) THEN GO TO QUIT ;

   /* NOW, DESCRIPTORS WITHIN A CATEGORY START */

ER6:    NER = 6 ;
        GET STRING (CF2) EDIT (N) ( F(2) ) ;
             SCAT (NCAT) = N ;

STEP3:  NDESC = NDESC + 1 ;
             PDESC (N) = NREC ;

R3:     READ FILE (CARDS) INTO (CARD) ;
             NREC = NREC + 1 ;  NC = NC + 1 ;
             IF ( CF2 = '  ' ) THEN GO TO R3 ;

        IF ( CF2 = '**' ) THEN GO TO STEP4 ;
 PAGE NUMBER :    3

 SOURCE CARD LISTING FOR
  P16:    /* DESCRIPTOR MASTER INDEX MAKEUP - FRI/14/JAN/77 */

        IF ( CF2 = '--' ) THEN GO TO STEP5 ;

ER6A:         NER = 6 ;
              GET STRING (CF2) EDIT (N) ( F(2) ) ;
                   GO TO STEP3 ;

   /* HERE, DONE WITH ONE CATEGORY - RECYCLING TO NEXT */

STEP4:  NER = 7 ;
ER7:    IF ( NCAT > 14 ) THEN GO TO QUIT ;

              ECAT (NCAT) = N ;
                   GO TO STEP2 ;

   /* NOW, DONE ONE (--) TYPE - RECYCLING TO NEXT */

STEP5:        ECAT (NCAT) = N ;
              DEND = NREC - 1 ;
                   GO TO STEP1 ;


/*****************************************/


EOFCARDS:     /* NOW, EOF ON INPUT DESCRIPTOR CARDS */
        MAXREC = NREC ;
        DEND = NREC ;
        ECAT (NCAT) = N ;

        CLOSE FILE (CARDS) ;

        OPEN FILE (CARDS) RECORD INPUT,
              FILE (MREF) RECORD UPDATE,
              FILE (DIND) RECORD OUTPUT ;
        ON ENDFILE (CARDS) GO TO EOJ ;

        READ FILE (MREF) INTO (MAST) ;
              REFS (4) = NREC ;
        REWRITE FILE (MREF) FROM (MAST) ;

   /* FIRST OUTPUT HEADER RECORD */

        CNUMS = 0 ;
        CNUMS(1) = MAXREC ;
        WRITE FILE (DIND) FROM (CARBIN) ;
        NOUT = 0 ;

   /* NOW, FOR THE COURSE INDEX */

        PCOV = ADDR (ICOURSE) ;
        PIN = ADDR (ICOURSE) ;

LP1:          DO J = 1 TO 4 ;
              CARD = COVER (J) ;
              WRITE FILE (DIND) FROM (CARD) ;
              END LP1 ;

              NOUT = NOUT + 4 ;

   /* VEHICLE INDEX IS OUTPUT HERE */

        PCOV = ADDR (IVEH) ;
        PIN = ADDR (IVEH) ;

LP2:          DO J = 1 TO 4 ;
              CARD = COVER (J) ;
              WRITE FILE (DIND) FROM (CARD) ;
              END LP2 ;

 PAGE NUMBER :    4

 SOURCE CARD LISTING FOR
  P16:     /* DESCRIPTOR MASTER INDEX MAKEUP - FRI/14/JAN/77 */
          NOUT = NOUT + 4 ;

  /* NOW FOR THE TASK INDEX OUTPUT */

      PCOV = ADDR (ITASK) ;
      PIN = ADDR (ITASK) ;

LP3:      DO J = 1 TO 4 ;
          CARD = COVER (J) ;
          WRITE FILE (DIND) FROM (CARD) ;
          END LP3 ;

          NOUT = NOUT + 4 ;

  /* NOW, OUTPUT THE DESCRIPTOR CARD FILE */

      DISPLAY ( '     ' ) ;
      DISPLAY ( 'DESCRIPTOR CARD FILE LISTING FOLLOWS' ) ;
      DISPLAY ( (36)'-' ) ;
      DISPLAY ( '     ' ) ;

      NCC = 0 ;

RALL: READ FILE (CARDS) INTO (CARD) ;
          NCC = NCC + 1 ;

      WRITE FILE (DIND) FROM (CARD) ;
          NOUT = NOUT + 1 ;

      IF ( CF2 = '--' ) THEN
          DO ;  DISPLAY ( '     ' ) ;  DISPLAY ( '    ' ) ; END ;

      IF ( CF2 = '**' ) THEN DISPLAY ( '     ' ) ;

      DISPLAY ( CARD ) ;

              GO TO RALL ;


/***************************************/


QUIT: /* THIS IS THE FORCED-TERMINATION ROUTINE */

      PUT STRING (DLINE) EDIT
          ( '** ERROR NUMBER ', NER, ' HAS OCCURED **' )
          ( A, F(2), A ) ;

      DISPLAY ( '     ' ) ;
      DISPLAY ( DLINE ) ;

      DLINE = EMSG ( NER ) ;

      DISPLAY ( DLINE ) ;
      DISPLAY ( '     ' ) ;

          GO TO EOJ ;


/***************************************/


/* FINALLY, EOF ON CARD FILE AGAIN */

EOJ:  CLOSE FILE (CARDS), FILE (DIND),
          FILE (MREF) ;

      DISPLAY ( '     ' ) ;
      PUT STRING (DLINE) EDIT
```

```
PAGE NUMBER :    5

SOURCE CARD LISTING FOR
 P16:     /* DESCRIPTOR MASTER INDEX MAKEUP - FRI/14/JAN/77 */

            ( '** TOTAL DESCRIPTOR INDEX CARDS READ : ',
            NCC, ' **' )  ( A, P'Z,ZZ9', A ) ;
       DISPLAY ( DLINE ) ;

       DISPLAY ( '      ' ) ;
       DISPLAY ( 'MASTER DESCRIPTOR INDEX PROCESSING '
            || 'IS NOW TERMINATING' ) ;
       DISPLAY ( '      ' ) ;

            END P16 ;
 PAGE NUMBER :    1

 SOURCE CARD LISTING FOR
  P17:    /* PROJECT FILE INITIALIZATION - WED/23/FEB/77 */

P17:    /* PROJECT FILE INITIALIZATION - WED/23/FEB/77 */
        PROC (INPARM) OPTIONS (MAIN) ;

   /*   REVISIONS :
        WED/23/FEB/77 - INITIAL IMPLEMENTATION.

        WED/02/MAR/77 - MODIFIED TO INITIALIZE ONLY CARD-IMAGE FILES
                        UP TO AND INCLUDING FILE ALTP.

        SUN/20/MAR/77 - MODS PRIOR TO TRANSMIT TO NCSS.
   */

DECLARE INPARM          CHAR (100) VARYING,
        PARM2           CHAR (4) BASED (PP) ;

DECLARE ( DISP, PTR, NPAGE, DPAGE, L, LT, NC,
             J, K )     FIXED BIN STATIC ;

DECLARE DTYPE (20)      CHAR (4) STATIC INIT (    /* FIRST 13 ARE CARD */
             'ID',   'REC',   'REJ',   'REV',     /* IMAGE TYPE        */
             'EXD',  'BPT',   'RKP',   'RRPJ',
             'RRPK', 'VQAL',  'VARF',  'SCEN',
             'ALTP',
             'REE',  'EXC',   'EXJ',   'EXV',     /* NOS 14 THRU 20,   */
             'CMR',  'TRER',  'FINR' ) ;          /* RECFM VARIES      */

DECLARE ( FLAGS(20), IFLAG ) BIT (1) STATIC,
        ( CARD, DLINE )       CHAR (80) STATIC,
        F4              CHAR (4)  DEF CARD,
        F2              CHAR (2)  DEF CARD,
        FN              CHAR (4)  DEF CARD POS (3),
        FNME            CHAR (8) STATIC,
        WAY (14:20)     LABEL ;

DECLARE  TTLS (20)      CHAR (40) STATIC INIT (
             'ID - PROJECT DESCRIPTION',
             'REC - COURSES ROE SEARCH ARGUMENTS',
             'REJ - TASKS ROE SEARCH ARGUMENTS',
             'REV - VEHICLES ROE SEARCH ARGUMENTS',
             'EXD - EXTRACT DEFAULTS',
             'BPT - BENEFIT PATTERN',
             'RKP - RISK PROFILE',
             'RRPJ - RISK REDUCTION PROJECTS',
             'RRPK - RISK REDUCTION PACKAGES',
             'VQAL - VARIABLES QUALITATIVE',
             'VARF - VARIABLE REFERENCES',
             'SCEN - SCENARIOS',
             'ALTP - ALTERNATE PROJECTS',
             'REE - ROE SEARCH RESULTS',
             'EXC - EXTRACT DB - COURSES',
             'EXJ - EXTRACT DB - JOBTASKS',
             'EXV - EXTRACT DB - VEHICLES',
             'CMR - COST MODEL RESULTS',
             'TRER - TREE RESULTS',
             'FINR - FINANCIAL RESULTS' ) ;

DECLARE T1              CHAR (18) STATIC INIT (
             'PAGE NUMBER : XXXX' ),

        T1PGE           PICTURE 'ZZZ9' DEF T1 POS (15),

        T2              CHAR (80) STATIC INIT (
            'LOAD OF PROJECT FILETYPE : XXXX' ),

        T2TTL           CHAR (53) DEF T2 POS (28) ;

/*************************************/

```

```
 PAGE NUMBER :    2

 SOURCE CARD LISTING FOR
  P17:     /* PROJECT FILE INITIALIZATION - WED/23/FEB/77 */

START:  FLAGS = '0'B ;
        NC, NPAGE, DPAGE,
           J, K = 0 ;
        PP = ADDR (INPARM) ;

        DISPLAY ( '     ' ) ;
        DISPLAY ( 'PROJECT FILE INITIALIZATION PROGRAM IS STARTING' ) ;
        DISPLAY ( '     ' ) ;

   /* HERE, DEVICE ASSIGNMENTS ARE VERIFIED */

FIRST:  DISP, PTR = 1 ;
        IF ( PARM2 = 'BOTH' ) THEN GO TO STEPA ;

        IF ( PARM2 = 'TERM' )
             THEN PTR = 2 ;
             ELSE DISP = 2 ;

   /* HERE, PRINTER DATASET IS INITIALIZED */

STEPA:  IF ( PTR = 2 ) THEN GO TO STEPB ;

        OPEN FILE (SYSPRINT)
             LINESIZE (120) PAGESIZE (58) ;

        ON ENDPAGE (SYSPRINT)
             BEGIN ;
             NPAGE = NPAGE + 1 ;

             PUT EDIT ( 'PAGE NUMBER : ', NPAGE, T2, '     ' )
                  ( PAGE, X(10), A, F(4),
                    SKIP(1), X(10), A, SKIP(1), A ) ;

             PUT SKIP (1) ;
             END ;

   /* HERE, INITIALIZATION OF TERMINAL DATASET PROCEEDS */

STEPB:  IF ( DISP = 2 ) THEN GO TO COMMENCE ;

        ON CONDITION (NEWDP)
             BEGIN ;
             IF ( LT < 61 ) THEN LT = 61 ;

             IF ( LT < 66 ) THEN
LPA:              DO L = LT TO 66 ;
                  DISPLAY ( '     ' ) ;
                  END LPA ;

             DISPLAY ( '     ' ) ;

             DPAGE = DPAGE + 1 ;
             T1PGE = DPAGE ;
             DISPLAY ( T1 ) ;
             DISPLAY ( '     ' ) ;
             DISPLAY ( T2 ) ;
             DISPLAY ( '     ' ) ;
             LT = 4 ;
             END ;

/****************************************/

COMMENCE: OPEN FILE (CARDS) RECORD INPUT ;
          ON ENDFILE (CARDS) GO TO PASS2 ;

        READ FILE (CARDS) INTO (CARD) ;
        NC = NC + 1 ;

        IFLAG = '0'B ;
```

```
PAGE NUMBER :    3

SOURCE CARD LISTING FOR
 P17:     /* PROJECT FILE INITIALIZATION - WED/23/FEB/77 */


        IF ( F4 = 'INIT' ) THEN
LP1:         DO ;
             IFLAG = '1'B ;
             READ FILE (CARDS) INTO (CARD) ;
             NC = NC + 1 ;
             END LP1 ;

        IF ( F2 = '**' ) THEN GO TO STEP2 ;

   /* HERE, FIRST CARD READ WAS NOT '**' CARD */

        T2TTL = '*** FIRST TITLE CARD IS MISSING ***' ;

STEP1:  IF ( PTR = 1 )
             THEN SIGNAL ENDPAGE (SYSPRINT) ;

        IF ( DISP = 1 )
             THEN SIGNAL CONDITION (NEWDP) ;
             ELSE DISPLAY ( T2 ) ;

R1:     READ FILE (CARDS) INTO (CARD) ;
        NC = NC + 1 ;

        IF ( F2 = '**' ) THEN GO TO STEP2 ;

        IF ( PTR = 1 ) THEN PUT EDIT (CARD) ( SKIP(1), X(10), A(80) ) ;

        IF ( DISP = 1 )
             THEN DISPLAY ( CARD ) ;

                  GO TO R1 ;

   /* HERE, A TITLE CARD HAS BEEN FOUND - VERIFY IT */

STEP2:       DO J = 1 TO 13 ;
             K = J ;
             IF ( FN = DTYPE(J) ) THEN GO TO STEP3 ;
             END STEP2 ;

        T2TTL = '*** TITLE CARD FILNAME -' || FN
             || '- IS INVALID ***' ;

                  GO TO STEP1 ;

   /* HERE, NEW TITLE (FILE) HEADER FOUND; VERIFY NON-DUPLICATE */

STEP3:  IF ( FLAGS(K) ) THEN
DUPFN:       DO ;
             T2TTL = '*** FILENAME - ' || FN || '- IS DUPLICATED ***' ;
                  GO TO STEP1 ;
             END DUPFN ;

        FLAGS (K) = '1'B ;
        T2TTL = TTLS (K) ;

        FNME = DTYPE (K) ;
        OPEN FILE (PROJ) RECORD OUTPUT TITLE ( FNME ) ;

        IF ( PTR = 1 ) THEN SIGNAL ENDPAGE (SYSPRINT) ;

        IF ( DISP = 1 ) THEN SIGNAL CONDITION (NEWDP) ;
             ELSE DISPLAY ( T2 ) ;

   /* HERE IS TEXT LINE OUTPUT LOOP */

R2:     READ FILE (CARDS) INTO (CARD) ;
        NC = NC + 1 ;
 PAGE NUMBER :   4

 SOURCE CARD LISTING FOR
  P17:    /* PROJECT FILE INITIALIZATION - WED/23/FEB/77 */
        IF ( F2 = '**' ) THEN
LP2:         DO ;
             CLOSE FILE (PROJ) ;
                  GO TO STEP2 ;
             END LP2 ;

        IF ( PTR = 1 )
             THEN PUT EDIT (CARD) ( SKIP(1), X(10), A(80) ) ;

        IF ( DISP = 1 )
             THEN DISPLAY ( CARD ) ;

        WRITE FILE (PROJ) FROM (CARD) ;

             GO TO R2 ;

/***************************************/

   /* HERE, EOF ON INPUT DATASET */

PASS2:  CLOSE FILE (PROJ), FILE (CARDS) ;

        IF ( ¬IFLAG ) THEN GO TO EOJ ;

        CARD = ' ' ;        /* DUMMY OUTPUT RECORD FOR FILE INIT */
        T2 = 'INITIALIZATION OF REMAINING FILES (IF ANY) WITH A '
             || 'SINGLE BLANK RECORD' ;

        IF ( PTR = 1 ) THEN SIGNAL ENDPAGE (SYSPRINT) ;
        IF ( DISP = 1 ) THEN SIGNAL CONDITION (NEWDP) ;

LP3:         DO J = 1 TO 13 ;    /* NOTE - ONLY INITIALIZING CARD
                                    IMAGE FILES UP THRU ALTP, INCL. */

             IF ( FLAGS(J) ) THEN GO TO LP3END ;

             OPEN FILE (PROJ) RECORD OUTPUT TITLE ( DTYPE(J) ) ;

             IF ( J <= 13 ) THEN GO TO WRITE80 ;
                            ELSE GO TO WAY (J) ;

WAY(14):
                  GO TO STEP4 ;
WAY(15):
                  GO TO STEP4 ;
WAY(16):
                  GO TO STEP4 ;
WAY(17):
                  GO TO STEP4 ;
WAY(18):
                  GO TO STEP4 ;
WAY(19):
                  GO TO STEP4 ;
WAY(20):
                  GO TO STEP4 ;

WRITE80:     WRITE FILE (PROJ) FROM (CARD) ;

STEP4:       CLOSE FILE (PROJ) ;
        DLINE = 'FILE : ' || DTYPE(J) ||
             ' HAS BEEN INITIALIZED WITH ONE (1) BLANK RECORD' ;

        IF ( PTR = 1 ) THEN PUT EDIT (DLINE) ( SKIP(2), X(10), A ) ;

        DISPLAY ( '     ' ) ;   DISPLAY ( DLINE ) ;

LP3END:      END LP3 ;

/***************************************/
```

```
PAGE NUMBER :    5

SOURCE CARD LISTING FOR
 P17:    /* PROJECT FILE INITIALIZATION - WED/23/FEB/77 */


   /* HERE IS THE END PROCESSING */

EOJ:     IF ( IFLAG ) THEN
LP4:          DO ;

              IF ( PTR = 1 ) THEN PUT EDIT
                     ( '** ALL DATASETS INITIALIZED **' )
                     ( SKIP(3), X(20), A ) ;

              DISPLAY (  '     ' ) ;
              DISPLAY ( '** ALL DATASETS INITIALIZED **' ) ;

              END LP4 ;

         IF ( PTR = 1 ) THEN PUT EDIT
              ( '** TOTAL NUMBER INPUT CARDS READ : ', NC, ' **' )
              ( SKIP(3), X(20), A, F(4), A ) ;

         PUT STRING (DLINE) EDIT
              ( '** TOTAL NUMBER INPUT CARDS READ : ', NC, ' **' )
              ( A, F(4), A ) ;
         DISPLAY ( '     ' ) ;   DISPLAY ( DLINE ) ;

         DISPLAY ( '     ' ) ;
         DISPLAY ( 'PROJECT FILE INITIALIZATION PROGRAM '
              || ' IS NOW TERMINATING' ) ;
         DISPLAY ( '     ' ) ;

              END P17 ;
```

```
PAGE NUMBER :   1

SOURCE CARD LISTING FOR
 P18:     /* INITILIZE MASTER REF FILE TO ZERO - WED/12/JAN/77 */


P18:      /* INITILIZE MASTER REF FILE TO ZERO - WED/12/JAN/77 */
          PROC OPTIONS (MAIN) ;

   /*     REVISIONS :
          WED/12/JAN/77 - FIRST IMPLEMENTEDED VERSION.

          MON/14/MAR/77 - MODS TO PREPARE FOR SHIP TO NCSS.
   */

DECLARE 1 MAST     STATIC,
            2 REFS (40)  FIXED BIN ;

          OPEN FILE (MREF) RECORD OUTPUT ;
          REFS = 0 ;
          WRITE FILE (MREF) FROM (MAST) ;
          CLOSE FILE (MREF) ;

                END P18 ;
 PAGE NUMBER :    1

 SOURCE CARD LISTING FOR
  P20:     /* PRINT ASSIGNED DESIGNATORS - WED/19/JAN/77 */

P20:    /* PRINT ASSIGNED DESIGNATORS - WED/19/JAN/77 */
        PROC (INPARM) OPTIONS (MAIN) ;

   /*   REVISIONS :
        WED/18/JAN/77 - STARTED WITH COURSE DESCRIPTORS ONLY.

        MON/31/JAN/77 - ADDING VEHICLE AND TASK VARIATIONS.

        SUN/20/MAR/77 - MODS PRIOR TO INSTALLATION AT NCSS.
   */

DECLARE INPARM    CHAR (100) VARYING,
        1 POVER   BASED (PP),
          2 PARM  CHAR (1),
          2 PARM2 CHAR (4),
        C1        CHAR (1) STATIC,
        C4        CHAR (4) STATIC,

        ( DIND, VEHS, TASKS )    FILE RECORD DIRECT
             KEYED ENV ( REGIONAL(1) ) ;

DECLARE DLINE     CHAR (80) STATIC,
        CFILL     CHAR (74) DEF DLINE POS (7),
        DFILL     CHAR (68) DEF DLINE POS (13),

        CARD      CHAR (80) STATIC,

        TTL1      CHAR (42) STATIC INIT (
             'PAGE XXXX : VEHICLE DESCRIPTOR ASSIGNMENTS' ),
        NPGE      PICTURE 'ZZZ9' DEF TTL1 POS (6),
        TYPE      CHAR (7) DEF TTL1 POS (13),

        ( DISP, PTR, LL, ND, NC, NPAGE, DPAGE, LT, NREC,
             MAXD, KMAX, J, CFLAG, KD, MAXD2,
             JD, L, LNOW, NER,
             K )       FIXED BIN STATIC,

        SRET      LABEL ;

/**************************************/

DECLARE 1 DINDEX       STATIC,        /* OVERALL DESCRIPTOR INDEX */
          2 DSTART     FIXED BIN,     /* CKF.DESC.INDEX */
          2 DEND       FIXED BIN,
          2 NCAT       FIXED BIN,
          2 NDESC      FIXED BIN,
          2 CATEGORIES (15),
            3 PCAT     FIXED BIN,
            3 SCAT     FIXED BIN,
            3 ECAT     FIXED BIN,
          2 PDESC (111) FIXED BIN,
        COVER (4)      CHAR (80) BASED (PCOV) ;

DECLARE 1 MAST         STATIC,
          2 REFS (40)  FIXED BIN,
        MREF           FILE RECORD ;

DECLARE 1 CINDESC      BASED (PTRCIN),     /* COURSE DESCRIPTORS */
          2 CINDS      CHAR (8),      /* CKF.CRS.DESC */
          2 FILL       CHAR (5),
          2 DTAB (100) FIXED BIN,

        CIN213    CHAR (213) STATIC ;

DECLARE 1 CINDER (304) STATIC,        /* CRS CIN DIRECTORY */
          2 CHCIN      CHAR (8),      /* CKF.CRS.DIRCIN */
          2 CINPTR     FIXED BIN,
        LOADCIN        CHAR (760)  BASED (PLCIN),
PAGE NUMBER :    2

SOURCE CARD LISTING FOR
 P20:     /* PRINT ASSIGNED DESIGNATORS - WED/19/JAN/77 */

          COVCIN (4)       CHAR (760)  BASED (PTCIN) ;

   DECLARE 1 VEHDESC       STATIC,          /* VEH DESCRIPTOR FILE */
               2 VEHDS     CHAR (13),       /* CKF.VEH.DESC */
               2 VTAB (100) FIXED BIN ;

   DECLARE 1 VEHDIR (200)  STATIC,          /* VEHICLE DIRECTORY */
               2 CHVEH     CHAR (13),       /* CKF.VEH.DIR */
               2 VEHPTR    FIXED BIN,
          COVVEH (4)       CHAR (750) BASED (PTVEH),
          LOADVEH          CHAR (750) BASED (PLVEH) ;

   DECLARE 1 TASKDESC      STATIC,          /* TASK DESCRIPTOR FILE */
               2 TASKDS    CHAR (13),       /* CKF.TASK.DESC */
               2 TTAB (100) FIXED BIN,

          1 TASKOVER       BASED (PTO),
              2 RATING     CHAR (5),
              2 RANK       CHAR (2),
              2 JOBTASK    CHAR (6) ;

   DECLARE 1 TASKDIR (1200) STATIC,         /* JOBTASK FILE DIRECTORY */
              2 CHTASK     CHAR (13),       /* CKF.TASK.DIR */
              2 TASKPTR    FIXED BIN,
          COVTASK (12)     CHAR (1500) BASED (PTTASK),
          LOADTASK         CHAR (1500) BASED (PLTASK) ;

   DECLARE 1 ATASKS        STATIC,          /* ABBR. TASK FILE */
              2 RATING     CHAR (7),        /* CKF.ATASKS */
              2 JOBTASK    CHAR (6),
              2 TASKTTL    CHAR (50),
              2 FILL       CHAR (6),
              2 BILCOST    PICTURE '(8)9V99',
              2 CC80       CHAR (1) ;

   DECLARE 1 AVEHICLES     STATIC,          /* ABBR. VEH FILE */
              2 STOCKN     CHAR (13),       /* CKF.AVEHS */
              2 DEVDESIG   CHAR (9),
              2 DEVNAME    CHAR (47),
              2 DEVCOST    PICTURE '(8)9V99',
              2 CC80       CHAR (1),

          VCOST            FIXED DEC (10,2) STATIC,
          VOVER            CHAR (80) BASED (PV80) ;

  /****************************************/

  START:  PP = ADDR ( INPARM ) ;
          PTCIN = ADDR (CINDER) ;
          PCOV = ADDR (DINDEX) ;
          PTO = ADDR (TASKDESC) ;

          PV80 = ADDR (AVEHICLES) ;
          PTVEH = ADDR (VEHDIR) ;
          PTTASK = ADDR (TASKDIR) ;

          PTRCIN = ADDR ( CIN213 ) ;

          NPAGE, DPAGE, NREC,
               K = 0 ;
          LT = 100 ;

 COMMENCE:
          OPEN FILE (MREF) RECORD INPUT,
               FILE (DIND) INPUT ;

          READ FILE (MREF) INTO (MAST) ;
               KMAX = REFS (4) ;
               MAXD = REFS (2) ;
```

```
PAGE NUMBER :    3

SOURCE CARD LISTING FOR
 P20:     /* PRINT ASSIGNED DESIGNATORS - WED/19/JAN/77 */


/*****************************************/

    /* HERE, COURSE DESCRIPTORS ARE SETUP */

TRYC:    IF ( PARM ¬= 'C' ) THEN GO TO TRYV ;

         TYPE = 'COURSE' ;
LP1:            DO J = 1 TO 4 ;
                READ FILE (DIND) INTO (CARD) KEY (J) ;
                COVER (J) = CARD ;
                END LP1 ;

         OPEN FILE (DESC) RECORD INPUT,
                FILE (DCIN) RECORD INPUT ;
         ON ENDFILE (DESC) GO TO EOJ ;

LP2:            DO J = 1 TO 4 ;
                PLCIN = ADDR (COVCIN(J)) ;
                READ FILE (DCIN) INTO (LOADCIN) ;
                END LP2 ;

                    GO TO CHECKDEV ;

    /* HERE, VEHICLE DESCRIPTORS ARE INITILIZED */

TRYV:    IF ( PARM ¬= 'V' ) THEN GO TO TRYT ;

         TYPE = 'VEHICLE' ;
         MAXD = REFS (8) ;
         PTRCIN = ADDR (VEHDESC) ;

         OPEN FILE (VDESC) RECORD INPUT,
                FILE (VEHS) INPUT,
                FILE (DVEH) RECORD INPUT ;
         ON ENDFILE (VDESC) GO TO EOJ ;

LP3:            DO J = 5 TO 8 ;
                READ FILE (DIND) INTO (CARD) KEY (J) ;
                COVER ( J - 4 ) = CARD ;
                END LP3 ;

LP4:            DO J = 1 TO 4 ;
                PLVEH = ADDR ( COVVEH(J) ) ;
                READ FILE (DVEH) INTO (LOADVEH) ;
                END LP4 ;

                    GO TO CHECKDEV ;

    /* HERE, SETUP OF TASK DESCRIPTORS */

TRYT:    IF ( PARM = 'T' ) THEN GO TO LP5 ;

ER1:     NER = 1 ;
         C1 = PARM ;  C4 = PARM2 ;
         PUT STRING (DLINE) DATA ( C1, C4 ) ;
                GO TO BUSTED ;

LP5:            DO J = 9 TO 12 ;
                READ FILE (DIND) INTO (CARD) KEY (J) ;
                COVER ( J - 8 ) = CARD ;
                END LP5 ;

         TYPE = 'JOBTASK' ;
         MAXD = REFS (12) ;
         MAXD2 = MAXD / 2 ;
         PTRCIN = ADDR ( TASKDESC ) ;

         OPEN FILE (TDESC) RECORD INPUT,
 PAGE NUMBER :    4

 SOURCE CARD LISTING FOR
  P20:     /* PRINT ASSIGNED DESIGNATORS - WED/19/JAN/77 */

                     FILE (TASKS) INPUT,
                     FILE (DTASK) RECORD INPUT ;
             ON ENDFILE (TDESC) GO TO EOJ ;

LP6:                 DO J = 1 TO 12 ;
                     PLTASK = ADDR ( COVTASK(J) ) ;
                     READ FILE (DTASK) INTO (LOADTASK) ;
                     END LP6 ;

     /* HERE, OUTPUT DEVICE ALLOCATION IS CHECKED */

CHECKDEV:
             DISP, PTR = 1 ;
             IF ( PARM2 = 'BOTH' ) THEN GO TO STEP1 ;

             IF ( PARM2 = 'TERM' )
                     THEN PTR = 2 ;
                     ELSE DISP = 2 ;

     /* HERE, PRINTER DATASET INITILIZED IF REQUIRED */

STEP1:       IF ( PTR = 2 ) THEN GO TO STEP2 ;

             OPEN FILE (SYSPRINT) STREAM PRINT
                     LINESIZE (120) PAGESIZE (60) ;

             ON ENDPAGE (SYSPRINT)
                     BEGIN ;
                     NPAGE = NPAGE + 1 ;
                     NPGE = NPAGE ;
                     PUT EDIT ( TTL1, '     ' )
                             ( PAGE, LINE(5), X(10), A(42), SKIP(1), A ) ;
                     PUT SKIP (1) ;
                     END ;

             SIGNAL ENDPAGE (SYSPRINT) ;

     /* HERE IS THE DISPLAY ENDPAGE ROUTINE, IF REQUIRED */

STEP2:       IF ( DISP = 2 ) THEN GO TO DLOOP ;

             ON CONDITION (NEWDP)
                     BEGIN ;
                     LNOW = LT ;
                     IF ( LT < 61 ) THEN LT = 61 ;

                     IF ( LT < 66 ) THEN
LP7:                         DO L = LNOW TO 66 ;
                             DISPLAY ( '     ' ) ;
                             END LP7 ;

                     DISPLAY ( '     ' ) ;
                     DPAGE = DPAGE + 1 ;
                     NPGE = DPAGE ;
                     DISPLAY ( TTL1 ) ;
                     DISPLAY ( '     ' ) ;
                     LT = 4 ;

             IF ( PARM = 'C' ) THEN DISPLAY
                     ( 'DESCRIPTORS ASSIGNED TO COURSES' ) ;
             IF ( PARM = 'V' ) THEN DISPLAY
                     ( 'DESCRIPTORS ASSIGNED TO VEHICLES' ) ;
             IF ( PARM = 'J' ) THEN DISPLAY
                     ( 'DESCRIPTORS ASSIGNED TO TASKS' ) ;
             DISPLAY ( '     ' ) ;
             LT = LT + 2 ;

                     END ;
 PAGE NUMBER :    5

 SOURCE CARD LISTING FOR
  P20:     /* PRINT ASSIGNED DESIGNATORS - WED/19/JAN/77 */
```

```
/************************************/

   /* THIS IS THE PROGRAM MAINLINE */

DLOOP:  IF ( PARM = 'T' ) THEN GO TO TLOOP ;
        IF ( PARM = 'V' ) THEN GO TO VLOOP ;

CLOOP:  READ FILE (DESC) INTO (CINDESC) ;
        NREC = NREC + 1 ;

        DLINE = 'CIN : ' || CINDS ;

LOOKCIN:     DO J = 1 TO MAXD ;
             IF ( CINDS = CHCIN(J) ) THEN GO TO STEP3 ;
             END LOOKCIN ;

        SUBSTR ( DLINE, 17 ) =
             '(NO MATCHING NITRAS DATA AVAILABLE)' ;

             GO TO STEP3 ;

   /* FOLLOWING IS THE VEHICLE FIRST LINE SETUP */

VLOOP:  READ FILE (VDESC) INTO (VEHDESC) ;
        NREC = NREC + 1 ;

        DLINE = VEHDS ;

LOOKV:       DO J = 1 TO MAXD ;
             K = J ;
             IF ( VEHDS = CHVEH(J) ) THEN GO TO FINDV ;
             END LOOKV ;

        SUBSTR ( DLINE, 15 ) =
             '/ ** NO MATCHING DATA IN VEHICLE FILE **' ;
                  GO TO STEP3 ;

FINDV:  READ FILE (VEHS) INTO (AVEHICLES) KEY (VEHPTR(K) ) ;
        SUBSTR ( DLINE, 15 ) = '/ ' || DEVDESIG
             || ' / ' || DEVNAME ;

                  GO TO STEP3 ;

   /* FOLLOWING IS SETUP OF FIRST LINE FOR TASKS */

TLOOP:  READ FILE (TDESC) INTO (TASKDESC) ;
        NREC = NREC + 1 ;

        DLINE = TASKOVER.RATING || ' / '
             || TASKOVER.RANK || ' / '
             || TASKOVER.JOBTASK ;

        IF ( TASKDS <= CHTASK(MAXD2) )
                THEN
             DO J = 1 TO MAXD2 ;
             K = J ;
             IF ( TASKDS = CHTASK(J) ) THEN GO TO FINDT ;
             END ;

                 ELSE
             DO J = MAXD2 TO MAXD ;
             K = J ;
             IF ( TASKDS = CHTASK(J) ) THEN GO TO FINDT ;
             END ;

       SUBSTR ( DLINE, 23 ) =
            '*** NO MATCHING DATA IN TASK FILE ***' ;
            GO TO STEP3 ;

```

```
 PAGE NUMBER :   6

 SOURCE CARD LISTING FOR
 P20:    /* PRINT ASSIGNED DESIGNATORS - WED/19/JAN/77 */
FINDT:  READ FILE (TASKS) INTO (ATASKS) KEY (TASKPTR(K)) ;

        SUBSTR ( DLINE, 23 ) = TASKTTL ;

             GO TO STEP3 ;

   /* HERE, FIRST LINE IS OUTPUT */

STEP3:  LL = 1 ;
             SRET = STEP4 ;
                  GO TO OUTLOOP ;

   /* NOW, BACK FROM FIRST LINE OUTPUT */

STEP4:  ND = 0 ;                    /* DTAB SS */
        CFLAG = 0 ;                 /* CATAGORY PRINT FLAG; 0 MEANS DO IT */
        NC = 1 ;                    /* CATEGORIES SS */

STEP5:  IF ( ND >= 100 ) THEN GO TO DLOOP ;

        ND = ND + 1 ;
        IF ( DTAB(ND) = 0 ) THEN GO TO STEP5 ;

        JD = DTAB (ND) ;    /* FOUND A NON-0 DESCRIPTOR */
        KD = PDESC (ND) ;   /* PTR TO DESCRIPTOR CARD */

        IF ( KD = JD ) THEN GO TO STEP6 ;
ER2:         NER = 2 ;
ER2A:        PUT STRING (DLINE) DATA ( ND, JD, KD, KMAX ) ;
                  GO TO BUSTED ;

STEP6:  IF ( KD <= KMAX ) THEN GO TO STEP7 ;
ER3:         NER = 3 ;
                  GO TO ER2A ;

STEP7:  IF ( ND >= SCAT(NC) ) THEN GO TO STEP9 ;

STEP8:       NC = NC + 1 ;
             CFLAG = 0 ;
             IF ( NC <= 15 ) THEN GO TO STEP7 ;
ER4:              NER = 4 ;
                       GO TO ER2A ;

STEP9:  IF ( ND = SCAT(NC) ) THEN GO TO STEP10 ;

        IF ( ND > ECAT(NC) ) THEN GO TO STEP8 ;

STEP10: IF ( CFLAG ¬= 0 ) THEN GO TO DETOUT ;

   /* HERE, CATAGORY LINE IS OUTPUT */

CATOUT: READ FILE (DIND) INTO (CARD) KEY ( PCAT(NC) ) ;
        DLINE = ' ' ;
        CFILL = CARD ;
        LL = 2 ;
        CFLAG = 1 ;
        SRET = DETOUT ;
             GO TO OUTLOOP ;

   /* HERE, DETAIL LINE IS OUTPUT */

DETOUT: READ FILE (DIND) INTO (CARD) KEY ( KD ) ;
        DLINE = ' ' ;
        DFILL = CARD ;
        LL = 3 ;
        SRET = STEP5 ;
             GO TO OUTLOOP ;

/*************************************/
```

```
 PAGE NUMBER :   7

 SOURCE CARD LISTING FOR
  P20:     /* PRINT ASSIGNED DESIGNATORS - WED/19/JAN/77 */

   /* THIS IS THE COMBINED DISPLAY / PRINTER OUTPUT ROUTINE */

OUTLOOP: IF ( DISP = 2 )              /* FIRST, TRY DISPLAY */
             THEN GO TO TRYPTR ;

         IF ( LL = 1 ) THEN
LINE1:       DO ;
             IF ( LT > 60 )
                  THEN DO ;
                       SIGNAL CONDITION (NEWDP) ;
                       END ;

                  ELSE DO ;
                       LT = LT + 3 ;
DB1:                   DISPLAY ( '   ' ) ;
DB2:                   DISPLAY ( '   ' ) ;
                       END ;

             GO TO DCCOUT ;
             END LINE1 ;

        IF ( LL = 2 ) THEN
LINE2:       DO ;
             IF ( LT > 61 )
                  THEN DO ;
                       SIGNAL CONDITION (NEWDP) ;
                            GO TO DCCOUT ;
                       END ;

                  ELSE DO ;
                       LT = LT + 2 ;
                            GO TO DB2 ;
                       END ;
             END LINE2 ;

LINE3:  IF ( LT > 62 ) THEN SIGNAL CONDITION (NEWDP) ;
                       ELSE LT = LT + 1 ;

DCCOUT: DISPLAY ( DLINE ) ;

TRYPTR: IF ( PTR = 2 )      /* PRINTER SERVICING ATTEMPTED HERE */
             THEN GO TO SRET ;   /* OTHERWISE, RETURN TO CALLER */

        IF ( LL < 3 )
             THEN IF ( LL = 1 ) THEN PUT SKIP (2) ;
                                ELSE PUT SKIP (1) ;

PTROUT: PUT EDIT ( DLINE ) ( SKIP(1), A(80) ) ;

             GO TO SRET ;   /* RETURN TO SUBROUTINE CALLER */

/***************************************/

   /* THIS IS THE ERROR PROCESSOR */

BUSTED: PUT STRING (CARD) LIST
             ( '      ** ERROR NUMBER ' || NER || ' HAS OCCURED **' ) ;

        IF ( DISP = 1 ) THEN
DCCERR:      DO ;
             DISPLAY ( '    ' ) ;
             DISPLAY ( '    ' ) ;
             DISPLAY ( CARD ) ;
             DISPLAY ( '    ' ) ;
             DISPLAY ( DLINE ) ;
             DISPLAY ( '    ' ) ;
             END DCCERR ;
 PAGE NUMBER :    8

 SOURCE CARD LISTING FOR
  P20:     /* PRINT ASSIGNED DESIGNATORS - WED/19/JAN/77 */

        IF ( PTR = 1 ) THEN
             PUT EDIT ( CARD, DLINE )
                  ( SKIP (3), A(80), SKIP(2), A(80) ) ;

/**************************************/

   /* THIS IS THE ENDFILE PROCESSOR */

EOJ:    PUT STRING (DLINE) EDIT
             ( '      ** RECORDS PROCESSED = ', NREC, ' **' )
             ( A, F(4), A ) ;

        IF ( DISP = 2 ) THEN GO TO STEP11 ;

             DISPLAY ( '    ' ) ;
             DISPLAY ( DLINE ) ;
             DISPLAY ( '    ' ) ;
             DISPLAY ( '    ' ) ;

STEP11: IF ( PTR = 1 ) THEN
             PUT EDIT ( DLINE) ( SKIP(3), A(80) ) ;

        CLOSE FILE (DIND), FILE (MREF),
             FILE (SYSPRINT) ;

             END P20 ;
```

APPENDIX E

AN INTERACTIVE COMPUTER INTERVIEW

This technique for probability encoding was reproduced from a report titled PROBABILITY ENCODING IN DECISION ANALYSIS* with the permission of one of its authors, Dr. Carl S. Spetzler.

---

* Spetzler, Carl S. and Staël von Holstein, Carl-Axel S., 1972. Probability Encoding in Decision Analysis, Paper presented at the ORSA-TIMS-AIEE 1972 Joint National Meeting, Atlantic City, N. J., 8-10 November 1972. Stanford Research Institute, Menlo Park, California.

Appendix B

AN INTERACTIVE COMPUTER INTERVIEW

B.1 Introduction

This appendix contains a sample session with the PEP program (Probability Encoding Program) developed by the Decision Analysis Group at Stanford Research Institute. This program is an interactive computer interview, which is based on the interval method for encoding. The subject is always given two or three intervals on which to bet and is asked to order the intervals in terms of his preferences. The program relies on an algorithm that finds the indifference point(s) by an iterative procedure. An opportunity always exists for the subject to bypass the iteration if he has decided on the indifference point. The program asks seven sets of questions, and the fractiles corresponding to the probability levels 1/6, 1/3, 1/2, 2/3, and 5/6 can be inferred from the answers. In fact, each fractile is encoded twice to provide a consistency check. The distribution is displayed in graphical and tabular form, and the subject is then made aware of any inconsistencies.

B.2 A Sample Session

The easiest way to demonstrate the program is by means of a sample session. The following example required roughly 25 minutes and cost around $9.00. The instructions part took up some of the time; an experienced user would find the length of the session to be between 10 and 15 minutes. All inputs by the subjects are underlined.

```
****************************************************************************
*  WELCOME TO THE DECISION ANALYSIS PROBABILITY ENCODING PROGRAM(PEP)      *
****************************************************************************


PLEASE TYPE YOUR NAME--CARL-AXEL STAEL VON HOLSTEIN
DO YOU NEED INSTRUCTIONS?      YES
THIS PROGRAM MEASURES YOUR PERSONAL JUDGEMENT ABOUT
AN UNCERTAIN QUANTITY.  YOUR JUDGEMENT IS THEN INTERPRETED
AS A PROBABILITY DISTRIBUTION.  KNOWLEDGE ABOUT PROBABILITY
WILL NOT NECESSARILY HELP YOU.  THERE ARE ALSO NO RIGHT
OR WRONG ANSWERS BECAUSE YOUR JUDGEMENT IS UNIQUE.

SO RELAX AND TRY TO RESPOND WITH YOUR JUDGEMENT.

MOST OF THE QUESTIONS WILL BE IN A FORM WHERE YOU NEED
TO MAKE DECISIONS.  YOU WILL BE GIVEN EITHER TWO OR
THREE NUMBERED RANGES.  YOU WILL BE ASKED TO DECIDE
WHICH OF THE RANGES ARE THE MOST LIKELY.

ONE WAY TO ANSWER THE QUESTIONS IS TO ASK YOURSELF:
"WHICH OF THESE RANGES WOULD I PREFER TO BET ON?"

YOU ARE EXPECTED TO ORDER THE RANGES IN TERMS OF YOUR
PREFERENCES.  FOR EXAMPLE, IF ONLY TWO RANGES ARE GIVEN,
YOU SIMPLY TYPE '12' OR '21'.  IF YOU ARE GIVEN THREE
RANGES AND IF YOU PREFER 3 TO 2 AND 2 TO 1, YOU TYPE
'321'.  IF YOU THINK 3 AND 2 ARE ABOUT EQUALLY LIKELY
BUT PREFER 3 AND 2 TO RANGE 1, TYPE '3=21'.

THE PROGRAM WILL ATTEMPT TO FIND RANGES THAT ARE EQUALLY
LIKELY IN YOUR JUDGEMENT.  WHEN YOU HAVE REACHED THIS
POINT TYPE 'EQ' OR '1=2' OR '1=2=3'.

YOU ARE ALSO ALLOWED A NUMBER OF OTHER RESPONSES.

DO YOU WISH TO SEE THE ENTIRE LIST?      YES

RESPONSES           MEANING
---------           -------

 12,21              YOU ARE COMPARING TWO NUMBERED RANGES.
                    YOU PREFER THE FIRST NUMBERED RANGE
                    TO THE SECOND.

123,321,ETC...      YOU ARE COMPARING THREE NUMBERED
                    RANGES. YOU PREFER THE FIRST
                    NUMBERED RANGE TO THE SECOND AND THE
                    SECOND TO THE THIRD, ETC...

1=23,2=31,ETC...    YOU ARE AGAIN COMPARING THREE NUM-
                    BERED RANGES. THE EQUAL SIGN INDI-
                    CATES EQUALLY LIKELY RANGES.
                    THE RANGES ARE STILL ORDERED BY
                    DESIRABILITY OR PREFERENCE.

EQ,EQUAL            IN YOUR JUDGEMENT THE RANGES ARE EQUALLY
                    LIKELY. '1=2=3' AND '1=2' ARE EQUIVALENT
                    TO TYPING 'EQ'.

BAC,BACK UP         YOU WISH TO CHANGE A PREVIOUS RESPONSE.
                    YOU MAY BACK UP A MAXIMUM OF FIVE LINES
                    BY REFERENCING A LINE NUMBER. YOU MAY
                    ALSO RETURN TO THE START OF A SECTION OF
                    QUESTIONS BY TYPING THE SECTION LETTER.
```

```
TER,TERMINATE        YOU WANT TO JUMP OVER THE REST OF THE
                     QUESTIONS AND SEE THE OUTPUTS.

SKI,SKIP             YOU KNOW WHAT VALUES DIVIDE THE RANGES.
                     THE PROGRAM WILL REQUEST THAT YOU TYPE
                     THEM IN DIRECTLY. SIMPLY TYPE IN
                     YOUR VALUES WHEN THE '?' APPEARS.

DIF,DIFFERENCE       THE PROGRAM PRINTS THE DIFFERENCES
                     BETWEEN THE RANGE BOUNDRARIES.

SUP,SUPPRESS         THE PROGRAM STOPS PRINTING DIFFERENCES.

QUI,QUIT             QUIT ENTIRELY. YOU WANT TO JUMP OVER
                     THE REST OF THE QUESTIONS AND THE OUTPUT


WITH THIS INFORMATION, GO AHEAD AND TRY THE PROGRAM.
ALL OF YOUR RESPONSES MUST BE TERMINATED BY PRESSING THE
RETURN KEY.  FEEL FREE TO ASK FOR MORE INFORMATION AT ANY
TIME.  JUST TYPE THE WORD 'HELP'.


WHAT VARIABLE IS TO BE INVESTIGATED ?     PROJECTED SALES

DOES PROJECTED SALES HAVE UPPER AND LOWER LIMITS ?     YES
WHAT IS THE UPPER LIMIT ?     500
WHAT IS THE LOWER LIMIT ?     100
```

[The program sets a limit to infinity if the limit is said not to exist.]

```
------------------**A**------------------
         RANGE1     RANGE2
LINE#  FROM    TO/FROM     TO

  1     100.     300.    500.     12

  2     100.     195.    500.     12

  3     100.     147.    500.     21

  4     100.     170.    500.     21

  5     100.     183.    500.     12

  6     100.     176.    500.     21

  7     100.     180.    500.     EQ
```

[180 is the 1/2 - fractile.]

```
-----------------**B**----------------
        RANGE1    RANGE2    RANGE3
LINE#  FROM   TO/FROM   TO/FROM    TO

  1     100.     139.     322.     500.      HELP

THREE RANGES WERE JUST PRINTED OUT.  YOU ARE EXPECTED
TO ORDER THESE THREE RANGES IN ACCORDANCE WITH THEIR
DESIRABILITY.
FOR EXAMPLE, IF YOU PREFER THE SECOND RANGE TO THE
FIRST RANGE AND THE FIRST RANGE TO THE THIRD, YOU
SHOULD RESPOND WITH '213'.  YOU ARE ALLOWED TO USE
AN EQUAL SIGN TO INDICATE EQUALLY LIKELY RANGES,
SUCH AS '12=3'.  THIS WOULD MEAN YOU CONSIDER THE
SECOND AND THIRD RANGES TO BE EQUALLY LIKELY,
BUT PREFER THE FIRST RANGE TO BOTH.

        RANGE1    RANGE2    RANGE3
LINE#  FROM   TO/FROM   TO/FROM    TO

  1     100.     139.     322.     500.      213

  2     100.     139.     248.     500.      213

  3     100.     139.     213.     500.      DIFFERENCES

  3     100.  (39.2)  139.  (73.8)  213.  (287.)  500.        231

  4     100.  (59.2)  159.  (53.8)  213.  (287.)  500.        1=23

  5     100.  (49.1)  149.  (47.0)  196.  (304.)  500.        21=3

  6     100.  (54.1)  154.  (33.7)  188.  (312.)  500.        321

  7     100.  (56.6)  157.  (35.3)  192.  (308.)  500.        EQ
```

[157 and 192 are the 1/3- and 2/3-fractiles, respectively.]

```
-----------------**C**----------------
             RANGE1              RANGE2
LINE#  FROM           TO/FROM             TO

  1     100.  (28.3)  128.  (28.3)  157.        SUPPRESS

  1     100.      128.      157.      12

  2     100.      114.      157.      21

  3     100.      121.      157.      21

  4     100.      125.      157.      1=2
```

[125 is the 1/6-fractile since it divides an interval with probability 1/3 into two equally likely intervals.]

```
------------------**D**------------------
           RANGE1    RANGE2
LINE#  FROM    TO/FROM     TO

  1    192.      346.     500.      BACK
WHERE TO ?     C
```

[The subject realized that his indifference point should be 127 for the range in Set C.]

```
------------------**C**------------------
           RANGE1    RANGE2
LINE#  FROM    TO/FROM     TO

  1     100.      128.     157.      SKIP

  1     100.  ?     127

  1     100.      127.     157.      EQ
------------------**D**------------------
           RANGE1    RANGE2
LINE#  FROM    TO/FROM     TO

  1     192.      346.     500.      12

  2     192.      265.     500.      SKI

  2     192.  ?     230

  2     192.      230.     500.      EQ
THIS ENDS THE FIRST QUESTION SET,A SHORTER SET FOLLOWS--PLEASE CONTINUE.
```

[230 is the 5/6 fractile. The first four sets of questions have established one value for each of the five fractiles. The next three sets are used for a consistency check. The subject can terminate the interview at this point (and anywhere else, too) and go directly to the outputs.]

```
-------------------**E**------------------
          RANGE1    RANGE2    RANGE3
LINE#  FROM   TO/FROM   TO/FROM    TO

  1     100.     121.     161.     180.      213

  2     100.     121.     151.     180.      BACK
WHERE TO ?       1

  1     100.     121.     161.     180.      231

  2     100.     131.     161.     180.      231

  3     100.     137.     161.     180.      12=3

  4     100.     134.     151.     180.      31=2

  5     100.     135.     156.     180.      1=2=3
```

[The whole range was assigned probability 1/2 in Set A. 135 and 156 are therefore the 1/6- and 1/3-fractiles.]

```
-------------------**F**------------------
          RANGE1    RANGE2
LINE#  FROM   TO/FROM     TO

  1     157.     174.     192.      21

  2     157.     183.     192.      12

  3     157.     179.     192.      EQ
```

[179 is a new value for the median (the 1/2-fractile) since the limits of the range corresponded to the probability levels 1/3 and 2/3 in Set B.]

```
DON'T GIVE UP--THIS IS THE LAST ONE
-------------------**G**------------------
          RANGE1    RANGE2    RANGE3
LINE#  FROM   TO/FROM   TO/FROM    TO

  1     180.     195.     341.     500.      SKI

  1     180.  ?      0210
WHAT ?
210

  1     180.     210.  ?      290

  1     180.     210.     290.     500.      EQ
```

[210 and 290 are new values for the 2/3- and 5/6-fractiles.)

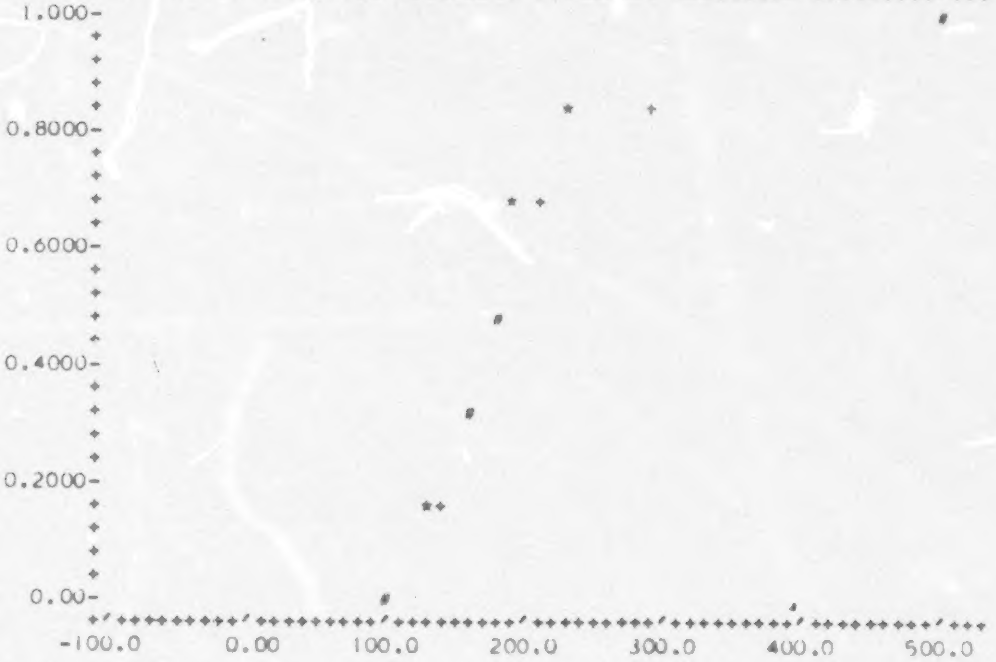
```
PLEASE ROLL THE PAPER FORWARD AND HIT CARRIAGE RETURN



*****************************************************************
*                         PEP : OUTPUT                          *
*****************************************************************

 VARIABLE NAME:PROJECTED SALES                       DATE 11/16/72

                     PREPARED BY CARL-AXEL STAEL VON HOLSTEIN
```



```
                 CUMULATIVE PROBABILITY TABLE
                 ----------------------------

   PROBABILITY                  THAT X IS LESS THAN
 ----------------               (****)       (++++)
      1.00                       500.         500.
      .833                       230.         290.
      .667                       192.         210.
      .500                       180.         179.
      .333                       157.         156.
      .167                       127.         135.
      .000                       100.         100.
```

[The subject can now see some inconsistencies, especially at the high end. The 5/6-fractile was first inferred to be 230 from Sed D and later to be 290 from Set F. The subject now has to reconcile the inconsistencies, but that is done outside of the program.]

APPENDIX F

BIBLIOGRAPHY

BIBLIOGRAPHY

Battersby, A., 1964. Network Analysis for Planning and Scheduling. Macmillan and Company Limited.

Baumol, W. J., 1965. Economic Theory and Operations Analysis. 2nd Edition. Englewood Cliffs, N.Y. Prentice-Hall.

Berners-Lee, C. M. (Editors) 1965. Models for Decision. London, E. C.: The English Universities Press Limited.

Bierman, Jr., H. and Smidt, S., 1971. The Capital Budgeting Decision. New York: The Macmillan Company, and London: Collier-Macmillan.

Braby, Richard Ed.D., Henry, James M., Parrish, William F., and Swope, William M. Ph.D., 1975. A Technique for Choosing Cost-Effective Instructional Delivery Systems. Training Analysis and Evaluation Group, Orlando, FL.

Chapman, Lid and J. P. Chapman, 1967. Genesis of Popular But Erroneous Psychodiagnostic Observations. Journal of Abnormal Psychology. Vol. 72, pp. 193-204.

De Greene, Kenyon B. (Ed), 1970. Systems Psychology. New York: McGraw-Hill Book Company.

Delphi Techniques and Cross-Impact Analysis, a Paper. Author unknown.

Eckenrode, R. T., 1965. Weighting Multiple Criteria. Management Sciences. Vol. 12, No. 3.

Edwards. W., 1961. Behavioral Decision Theory. Annual Review of Psychology. 12, 473-498.

Edwards, W., 1963. Probabilistic Information Processing by Man, Machines and Man-Machine Systems. Santa Monica, Calif. System Development Corp. TM-1418/000/01., August.

Edwards, W., 1975. Research on the Technology Inference and Decision. Final Technical Report. SSRI Report 75-10.

Fisher, G. W., 1972. Four Methods for Assessing Multi-Attribute Utilities: An Experimental Validation. ONR Engineering Psychology Programs. NR197-014.

Fishburn, P. C., 1966. Decision Under Uncertainty: An Introductory Exposition. Journal of Industrial Engineering 27(7).

Fishburn, P. C., 1967. Methods of Estimating Additive Utilities. Management Sciences. Vol. 13 No. 7.

Fleishman, E. A., 1975. "Taxonomic Problems in Human Performance Research." In Singleton, W. T. and Spurgeon, P. (Editors), Measurement of Human Resources London: Taylor and Francis.

Freedy, A., et al., 1974. Adaptive Computer Aiding in Dynamic Decision Processes Part I. Adaptive Decision Models and Dynamic Utility Estimation. Technical Report PTR-1016-74-5 (I), Perceptronics, Inc., Prepared for the Office of Naval Research under Contract N00014-73-C-0286.

Gagné, R. M., 1970. Conditions of Learning. (2nd Ed.) New York: Holt Rinehart, Winston.

Gagné, R. M., 1975. "Taxonomic Problems of Educational Systems." In Singleton, W. T. and Spurgeon, P. (Editors), Measurement of Human Resources. London: Taylor and Francis.

Gagné, R. M. and Briggs, L. J., 1974. Principles of Instructional Design. New York: Holt, Rinehart and Winston.

Giauque, W. C., 1975. Decision Making in the Naval Education and Training System, TAEG Report No. 27, Training Analysis and Evaluation Group, Orlando, FL.

Goldberg, L. R., 1968. Simple Models or Simple Processes? American Psychologist. Vol. 23, pp 483-496.

Guilford, J. P., 1954. Psychometric Methods, N. Y. McGraw-Hill.

Gustafson, D. H., G. K. Pai and G. C. Kramer, 1971. A "Weighted Aggregate" Approach to R & D Selection, AIIE Transactions, Vol. III, No. 1.

Hahn, Gerald J. and Shapiro, Samuel S., 1967. Statistical Models in Engineering. New York: John Wiley & Sons, Inc.

Halpin, S. M. and J. A. Thornberry and S. Streufert, 1973. The Credibility of Computer Estimates in a Simple Decision Making Task. ONR Technical Report #5. ONR Organizational Effectiveness Research Programs, Code 2452.

Handbook for Decision Analysis, 1973. A user-oriented document prepared by Decision and Design, Inc., for the Advanced Research Projects Agency and Office of Naval Research under Contract N00014-73C-0149.

Huber, G. P., 1974. Methods for Quantifying Subjective Probabilities and Multi-Attribute Utilities. The Journal for the American Institute for Decision Sciences. Vol. 5, No. 3.

Huber, G. P., 1974. Multi-Attribute Utility Models: A Review of Field and Field-Like Studies. Management Sciences. Vol. 20, No. 10. June.

Kahneman, D. and A. Tversky, 1973. On the Psychology of Prediction. Psychological Review. Vol. 80, No. 4. (July).

Miller, G. A., 1956. The Magical Number Seven Plus or Minus Two: Some Limits on Our Capacity for Processing Information. Psychological Review Vol. 63, 81-97.

Miller, Robert B. and Larry R. Duffy, 1975. Design of Training Systems, Phase IIA Final Report, TAEG Report No. 12-3. Training Analysis and Evaluation Group, Orlando, FL.

Miller, Robert B., and Duffy, Larry R., 1975. Design of Training Systems, The Development of Scaling Techniques, TAEG Report No. 32, Training Analysis and Evaluation Group, Orlando, FL.

Miller, R. B., 1953. Handbook on Training and Training Equipment Design. Pittsburgh, Pennsylvania: American Institute for Research, Wright Air Development Center Technical Report 53-136.

Miller, R. B., 1962. "Task Description and Analysis." In Gagne, R. M. (Editor), Psychological Principles in System Development. New York: Holt Rinehart, Winston.

Miller, R. B., 1967. "Task Taxonomy: Science or Technology?" In Singleton, W. T., Easterly, R., Whitfield, D., (Editors), The Human Operator in Complex Systems. London: Taylor and Francis.

Miller, R. B., 1971. Development of a Taxonomy of Human Performance: Design of a Systems Task Vocabulary. Washington, D.C.: American Institutes for Research and U. S. Army Behavior and Systems Research Laboratory.

Miller, R. B., 1971. Development of a Taxonomy of Human Performance: A User-Oriented Approach. Washington, D.C.: American Institutes for Research and U. S. Army Behavior and Systems Research Laboratory.

Miller, R. B., 1972. Theoretical Background to the Design of Duty Modules. Washington, D.C.: American Institutes for Research and U. S. Army Behavior and Systems Research Laboratory.

Miller, R. B., 1974. A Method for Determining Task Strategies. Brooks Air Force Base, Texas: American Institutes for Research and Air Force Human Resources Laboratory.

Miller, R. B., 1975. "Taxonomies for Training." In Singleton, W. T., and Spurgeon, P., (Editors), Measurement of Human Resources. London: Francis and Taylor.

Miller, Robert B., and Smode, Alfred F., 1976. Major Innovations in Training Technology, Training Analysis and Evaluation Group, Orlando, FL.

Murphy, A. H. and R. L. Winkler, 1973. Subjective Probability Forecasting of Temperature: Some Experimental Results. Proceedings of the Third Conference on Statistics in Atmospheric Science. Amer. Meteorological Society. Boulder, Colorado, June.

Peterson, C. R., 1971. Judgments of Probability and Utility for Decision Making. First Annual Report. DMR Contract No. N00014-67-A-0181-0034. NR 197-014.

Peterson, C. and L. Beach, 1967. Man as an Intuitive Statistician. Psychological Bulletin, July, 68 (1), 29-46.

Rundquist, E. A., 1971. Job Training Course Design and Improvement. San Diego, California: Navy Personnel and Training Research Laboratory Research Report SRR 71-4 (2nd Edition).

Samers, B. N., Dunham, A. D., and Nordhauser, F., 1974. The Development of a Methodology for Estimating the Cost of Air Force On-the-Job Training. Report No. AFHRL-TR-74-34, Contract F41609-72-C-0048, Hq. Air Force Human Resources Laboratory, Brooks, AFB, Texas.

Schlaifer, R., 1959. Probability and Statistics for Business Decisions-An Introduction to Managerial Economics Under Uncertainty. New York: McGraw-Hill Company.

Schlaiffer, R., 1967. Analysis of Decisions Under Uncertainty, Vol. 1. N. Y. McGraw-Hill.

Schum, D. A., 1970. Behavioral Decision Theory and Man-Machine Systems. In DeGreene (Ed.) Systems Psychology, N. Y. McGraw-Hill.

Singer, R. N. (Ed), 1972. Readings in Motor Learning. Philadelphia: Lea and Fabinger.

Slovic, P., 1966. Value as a Determiner of Subjective Probability. IEEE Transactions on Human Factors in Electronics, March, HFE-7, 22-28.

Slovic, P. and S. Lichtenstein, Nov. 1971, Comparison of Bayesian and Regression Approaches to the Study of Information Processing Judgment. Organizational Behavior and Human Performance, Vol. 6, No. 6, pp. 649-744.

Spetzler, C. S. and C. S. Staël von Holstein, 1972. Probability Encoding in Decision Analysis. Paper presented at the ORSA-TIMS-AIEE Joint National Meeting, Atlantic City, N.J. (November).

Staël von Holstein, C. S., 1974. Procedures for Encoding Probability Distributions from Groups. Menlo Park, Calif. Stanford Research Institute Project Memorandum, SRI Project 2753-2, May.

Van Horne, James C., 1968. Financial Management and Policy. New Jersey: Prentice-Hall, Inc.

v. Winterfeldt, D., 1971. Multi-Attribute Utility Theory: Theoretical Background and an Experimental Validation. Diploma Thesis, University of Hamburg (Germany).

v. Winterfeldt, D. and G. W. Fischer, 1973. Multi-Attribute Utility Theory: Models and Assessment Procedures. Invited review paper presented at the Fourth Conference on Subjective Probability, Utility and Decision Making, Rome, Sept. 3-6.

v. Winterfeldt, D., 1975. An Overview, Integration and Evaluation of Utility Theory for Decision Analysis, Technical Report, sponsored by DARPA, SSRI Research Report 75-9.

Wasson, C. R., 1965. The Economics of Managerial Decision: Profit Opportunity Analysis. New York: Appleton-Century-Crofts.

DISTRIBUTION LIST

Air Force

Headquarters, Air Training Command (XPTD, Dr. D. E. Meyer)
Headquarters, Air Training Command (XPTIA, Mr. Goldman)
Air Force Human Resources Laboratory, Brooks Air Force Base
Air Force Human Resources Laboratory (Library), Lowry Air Force Base
Air Force Office of Scientific Research/AR (Dr. A. R. Fregly)

Army

Commandant, TRADOC (Technical Library)
Army Research Institute (Dr. Ralph R. Canter, 316C; Dr. Edgar Johnson)

Coast Guard

Commandant, U.S. Coast Guard Headquarters (G-P-1/62; G-RT/81)

Marine Corps

CMC (Code OT)
CGMCDEC (Mr. Greenup)
Director, Marine Corps Institute

Navy

Assistant Secretary of the Navy (R&D) (4E741, Dr. S. Koslov)
CNO (OP-987P7, CAPT Connery; OP-991B, M. Malehorn; OP-987F10, Dr. R. Smith;
    OP-987, H. Stone)
NAVCOMPT (Code NCD-7)
ONR (Code 458, 455)
ONRBO Boston (J. Lester)
ONRBO Chicago
ONRBO Pasadena (E. E. Gloye)
CNM (MAT-03424, Mr. A. L. Rubinstein)
CNET (01, 00A, N-5 (6 copies))
CNAVRES (Code 02)
COMNAVSEASYSCOM (03, 047C1, 047C12)
COMNAVAIRSYSCOM (03, 340F)
CNET SUPPORT (00, 01A)
CNTECHTRA (0161, Dr. Kerr (5 copies))
CNATRA (F. Schufletowski)
COMTRALANT
COMTRALANT (Educational Advisor)
COMTRAPAC

DISTRIBUTION LIST (continued)

Navy (continued)

CO NAVPERSRANDCEN (Code 02, Dr. Regan; Dr. Earl Jones; Library)
NAVPERSRANDCEN Liaison (Code 01H)
Superintendent NAVPGSCOL (Code 2124)
Superintendent Naval Academy Annapolis (Chairman, Behavioral Science Dept.)
CO NAVEDTRAPRODEVCEN (AH3)
CO NAVEDTRASUPPCEN NORVA
CO NAVEDTRASUPPCENPAC (5 copies)
CO NAVAEROMEDRSCHLAB (Chief Aviation Psych. Div.)
CO FLTCOMDIRSYSTRACENPAC
CO NAMTRAGRU
CISO, NTTC Corry Station
CO NAVTRAEQUIPCEN (N-21, N-215, N-131 (2 copies), N-2211, N-00AF, N-00M, PM TRADE)
Center for Naval Analyses (2 copies)
U.S. Naval Institute (CDR Bowler)

Other DOD

Military Assistant for Human Resources, OAD (E&LS) ODDR&E (COL Henry Taylor)
OASD (I&L) - WR (LT COL Grossel)
Director, Human Resources Office, Defense Advanced Research Projects Agency
(R. Young)
Defense Advanced Research Projects Agency (CTO, Dr. H. F. O'Neil, Jr.)
Institute for Defense Analyses (Dr. Jesse Orlansky)

Non-DOD

Essex Corporation (Dr. John Collins)

Information Exchanges

DDC (12 copies)
DLSIE (James Dowling)
Scientific Technical Information Office, NASA
Executive Editor, Psychological Abstracts, American Psychological Association
ERIC Processing and Reference Facility, Bethesda, MD (2 copies)